World Wide Web (2012) 15:603–630
DOI 10.1007/s11280-011-0146-0

Exploring latent browsing graph for question answering recommendation

Meng-Fen Chiang · Wen-Chih Peng · Philip S. Yu

Received: 15 April 2011 / Revised: 24 September 2011 /
Accepted: 10 October 2011 / Published online: 25 October 2011

Abstract In this paper, we develop a framework of Question Answering Pages (referred to as QA pages) recommendation. Our proposed framework consists of the two modules: the off-line module to determine the importance of QA pages and the on-line module for on-line QA page recommendation. In the off-line module, we claim that the importance of QA pages could be discovered from user click streams. If the QA pages are of higher importance, many users will click and spend their time on these QA pages. Moreover, the relevant relationships among QA pages are captured by the browsing behavior on these QA pages. As such, we exploit user click streams to model the browsing behavior among QA pages as QA browsing graph structures. The importance of QA pages is derived from our proposed QA browsing graph structures. However, we observe that the QA browsing graph is sparse and that most of the QA pages do not link to other QA pages. This is referred to as a *sparsity problem*. To overcome this problem, we utilize the latent browsing relations among QA pages to build a *QA Latent Browsing Graph*. In light of QA latent browsing graph, the importance score of QA pages (referred to as Latent Browsing Rank) and the relevance score of QA pages (referred to as Latent Browsing Recommendation Rank) are proposed. These scores demonstrate the use of a QA latent browsing graph not only to determine the importance of QA pages but also to recommend QA pages. We conducted extensive empirical experiments on Yahoo! Asia Knowledge Plus to evaluate our proposed framework.

M.-F. Chiang · W.-C. Peng (✉)
National Chiao Tung University, Hsinchu, Taiwan
e-mail: wcpeng@g2.nctu.edu.tw

M.-F. Chiang
e-mail: mfchiang.cs95g@nctu.edu.tw

P. S. Yu
University of Illinois at Chicago, Chicago, IL 60607, USA
e-mail: psyu@cs.uic.edu

Keywords browsing graph · question answering · recommendation

1 Introduction

Question and Answering (QA) forums, such as Baidu and Yahoo! Answers, are effective tools in searching for information and knowledge on the Internet. In general, a QA forum features a portal in which users form several communities and contribute their questions and answers. The contributed information and knowledge is a rich repository to fulfill the information needs of users by providing an on-line QA search engine or QA recommendation system.

Most QA forums, such as Yahoo! Asia Knowledge Plus, provide QA search or QA recommendation services based on a textual model that exploits the textual similarity. By issuing keywords, users could get a ranked list of QA pages that contain some words similar to the keywords issued. From a ranked list of QA search results, users can either sort through the QA pages of interest or issue other keywords if the current search results do not satisfy their requirements. Without loss of generality, most QA portals will provide a ranked list of relevant QA pages when a QA page (referring to as a query QA page Q) is browsed. For example, for each browsed QA page in Yahoo! Asia Knowledge Plus, a ranked list of relevant QA pages are also displayed in the current QA page to fulfill the information needs of users if users want to see more relevant QA pages after browsing the current QA page. The existing solutions for relevant QA recommendations in Yahoo! Asia Knowledge Plus use a textual model to retrieve those QA pages that contain the same keywords as the query QA page, Q. However, one of the disadvantages of the textual models is that, even though QA pages contain the same keyword as in the query QA page, their topics may be irrelevant to the topic of the query QA page. Note that the textual models are language-dependent. Consequently, a textual model that is designed for a particular language may fail for another language. To overcome the above disadvantages of using textual models, we explore the actual browsing behavior of users to determine how important and relevant of QA pages.

Another characteristic of QA forums is the variation in quality. When users contribute information to QA forums, the quality of QA pages may vary. The accurate evaluation of the quality of QA pages is essential under these circumstances. A number of QA forums enable users to rate QA pages to obtain a high quality of user-generated content. The rating information provided by users is then used to filter out QA pages of a lower quality. Several studies have elaborated on designing a quality-aware search on QA forums [1, 3, 4, 7, 20] with user-generated content (for example, rating information). For example, the authors in [3, 4] proposed a semi-supervised approach for retrieving relevant and high quality content in QA forums. The authors in [3, 4] modeled user reputations and used less manual supervision to retrieve relevant and high quality answers by integrating content quality and user reputation information into their ranking process. The authors in [20] introduced a quality-aware QA framework that considered both answer relevance and quality in selecting the answers to be returned. Several methods have been developed for estimating content quality by considering the expertise of a user in both asking and answering questions. However, most prior studies on estimating answer quality are limited in analyzing language-dependent user-generated content. Moreover, user-generated content, such as rating information, may be manipulated and biased and thus cannot accurately represent the users reactions to the content of QA pages. As indicated in [5], over 30% of the best answer selections in Yahoo! Answers are affected by the users who provide the answers.

We claim that the importance of QA pages and the relevance of QA pages for QA recommendation by investigating and modeling the user browsing logs is a more reliable method because the user-generated content is unreliable and the textual models are language-dependent. Without a loss of generality, the user browsing logs record information about when and which Web pages are clicked on, and by whom. Note that the user browsing logs record the time-stamp, user identity, and URL information. To facilitate our presentation, Web pages that are not QA pages are referred to as *non-QA pages* because user browsing logs consist of not only QA pages but also a number of non-QA pages. Several types of relations among QA pages were identified from the user browsing logs. In this paper, the user browsing logs are modeled as a QA browsing graph in which each node represents one QA page and the edges represent the browsing behaviors of users. Then, we adopt the state-of-the-art ranking approaches to compute both the importance and relevance scores for QA pages. For example, we adopted the BrowseRank algorithm in our QA browsing graph to determine the importance of QA pages. However, a naive QA browsing graph that only considers explicit relations among Web pages is too sparse to link the QA pages. According to our observation from Yahoo! Answers, approximately 54% of QA pages are isolated in a QA browsing graph. As illustrated in Figure 1, the link distribution reveals the notable sparsity problem that is faced by naive modeling of the user browsing logs. Consequently, the naive adoption of BrowseRank fails to determine the importance scores for isolated QA pages.

To address the sparsity problem, we explore the latent browsing relations among QA pages to build a *QA Latent Browsing Graph*. In particular, we adopt a time-dependent Markov property to determine whether two QA pages are context-dependent or not. The QA pages are context-dependent or latently related to a previously visited QA page if their time difference is not larger than a given time constraint. The time constraint is used to explore the latent browsing behavior of QA pages. However, the setting of this time constraint is an important issue. To derive a time constraint, we propose a behavioral coherence evaluation approach to evaluate the quality of a time constraint for each QA page. Once a QA latent browsing

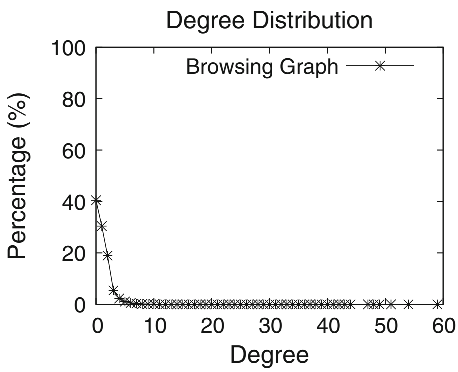


Figure 1 Degree distribution from Yahoo! Asia Knowledge Plus (July 15, 2009–July 17, 2009).

graph is built for QA pages on user behavior logs, we determine the staying time distributions and compute their latent browsing rank of the given QA pages. By using the QA latent browsing graph, we propose a Latent Browsing Rank (abbreviated as LBR) of QA pages to determine the importance of QA pages but also recommended QA pages with a higher Latent Browsing Recommendation Rank (abbreviated as LBRR). We develop a framework that consists of the off-line module and the on-line module. The QA latent browsing graph is first built and the LBR values of QA pages are determined in the off-line module. A recommendation list of QA pages that are ranked by LBRR is generated in the on-line module. We conducted extensive experiments to demonstrate the effectiveness of latent browsing relations on the real data set, that is, Yahoo! Asia Knowledge Plus. The experimental results indicate that our framework is able to determine high quality QA pages and recommend relevant QA pages, which demonstrates the advantage of exploring the QA latent browsing graph.

In summary, the main contributions of this study are as follows:

- We propose a QA latent browsing graph to capture the user browsing behavior in QA forums.
- We utilize the QA latent browsing graph and staying time information to determine the importance score of QA pages (i.e., LBR).
- We exploit Random Walk with Restart in the QA latent browsing graph to derive the relevance score of QA pages (i.e., LBRR).
- We conducted experiments on a large-scale real data set (i.e., Yahoo!Asia Knowledge Plus) to evaluate our proposed QA latent browsing graph and algorithms.

The remainder of this paper is organized as follows. Section 2 provides an overview of the related work. The background information of this study is described in Section 3. Section 4 introduces our observations and presents the QA latent browsing graph in modeling the user-perceived relevances. In Section 5, we analyze latent browsing relations. Section 6 describes two algorithms to determine LBR and LBRR of QA pages. Section 7 presents the performance studies. Section 8 concludes this paper.

2 Related work

We briefly review previous studies on link-based ranking algorithms in this section. Then, we describe the state-of-the-art research works of Question Answering Systems.

2.1 Link-based ranking algorithm

Numerous studies have focused on exploring the linkage relationships among data items for ranking [8, 13]. A typical example is to model Web data as a link graph, where the nodes represent Web pages and the edges represent the hyper-links. Then, a link-based ranking algorithm (e.g., PageRank) is used to determine the importance of pages. In principle, PageRank considers Web pages as more important if they are pointed by more links from more important pages. Specifically, PageRank simulates the random walk of a "Web surfer" on the graph, and the importance score is defined as the stationary probability of the discrete-time Markov process.

Several algorithms were developed to improve the performance of PageRank [9, 11]. For example, [9] discussed several possible alternatives to enhance the basic model of PageRank, such as storage issues, convergence properties, and updating problems. In contrast to the static network in traditional PageRank, the authors in [11] proposed BrowseRank that explores the dynamic hyper-link transitions to model user behavior data. The basic idea in BrowseRank is to formulate a browsing graph based on the browsed hyper-links, in which each vertex represents a Web page and the edges represent the browsed hyper-link transitions between Web pages. Furthermore, a staying time distribution is determined for each Web page from the observed staying time information. The importance score was affected not only by the underlying linkage structure but also by the staying time distribution of all Web pages. In principle, similar to PageRank, BrowseRank considers Web pages as more important if they are linked by more links from more important pages. The higher the ratios of time that "Web surfers" spend on a particular page to the time they spend on all of the pages, the more likely it is that the page is important.

Another variation of PageRank is the computation of the reachability for a query node. For example, Random Walk with Restart analyzes the reachability of a particular query node to the remaining destination nodes. The basic idea of Random Walk with Restart is as follows: the importance information is propagated by two ways: (1). jump back to the query node with probability c and (2). propagate to their adjacent neighbors with probability (1-c). Consequently, given a query node, the more paths that connect a destination node to the query node within a few hops, the more likely it is that the destination node is relevant. Random Walk with Restart was proven to be successful in several applications, such as in content-based image retrieval [6], cross modal correlation discovery [15, 19], and a movie recommender system [12].

2.2 A quality-aware question answering system

A considerable amount of research efforts has been dedicated to user preference mining [10] and Web content filtering [2, 14, 16]. Question and Answering (QA) forums, such as Baidu and Yahoo! Answers, are essential among various Web contents in searching for information and knowledge on the Internet. Although the question-answering (QA) systems are a valuable repository for user-generated content, the distribution of content quality exhibits a high variance. Several algorithms for content quality estimation in QA systems were developed to enhance further applications [1, 3, 4, 7, 20].

In [7], a stochastic model was built from manually labeled data to predict the quality of a question and answer pair (QA) by determining the correlations among non-textual answer features and answer quality. The set of non-textual answer features includes answer length, the number of answers of the respondent, the current questions and best answers, and answer rating. In [1], multiple features, such as textual relevant features, user interaction features and content usage statistics, were used to estimate the quality of the QA content. The authors in [3, 4] recently modeled the user reputations and used less manual supervision to retrieve relevant and high quality answers by integrating content quality and user reputation information into

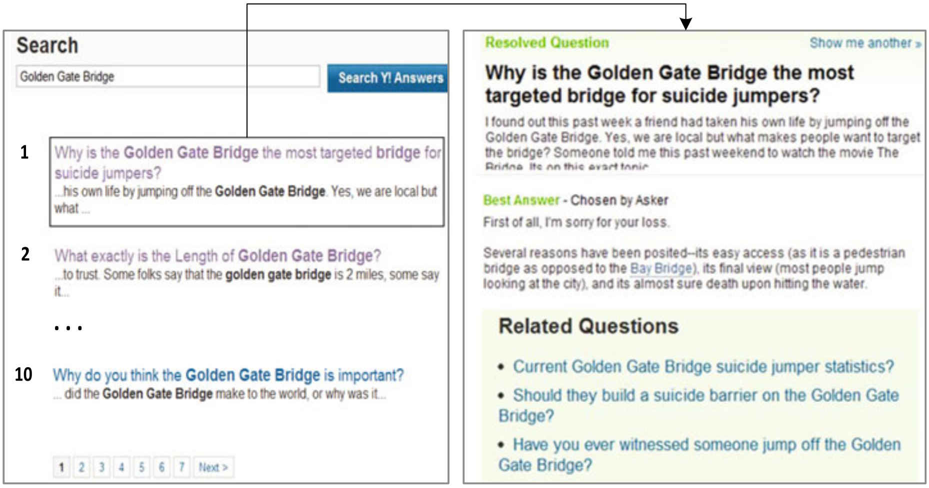


Figure 2 An example of non-QA page and QA page.

the ranking process. The authors in [20] introduced a quality-aware QA framework that considered both the answer relevance and the quality in selecting the answers to be returned. Several methods were developed to estimate the content quality by considering the expertise of a user in both asking and answering questions. However, most of the prior studies on estimating answer quality are confined to user-generated content. To the best of our knowledge, there is no prior work on computing the importance scores of QA pages or determining relevance scores between QA pages from user behavior data. The large amount of daily user behavior data contains valuable information, which motivates the development of the model and the algorithms to compute the importance scores for QA pages and to determine the relevance scores among QA pages.

3 Preliminaries

In the domain of Question and Answering (QA) forums, two types of Web pages are considered: QA pages and non-QA pages. Figure 2 shows an example of a non-QA page (left) and an example of a QA page (right) from Yahoo! Answers.[1] In Figure 2, the non-QA page is a search result page that contains a list of QA pages after a keyword "Golden Gate Bridge," is issued. The figure on the right side of Figure 2 is an example of a QA page when a user clicked on a search result from the figure of the left side of Figure 2. As illustrated in Figure 2, a QA page contains three types of content: (1) question content: the content of a posted question, (2) answer content: the content of a set of answers, and (3) recommendation list: a ranked list of hyperlinks that correspond to the QA pages returned by the current QA recommendation service.

[1] Some personalized information is removed for privacy issue.

Table 1 A snippet of a user's click stream.

Date	April 23								April 24							
Visiting time	0	10	20	25	30	35	40	50	0	10	20	25	35	100	110	120
Visiting page (non-QA)	P_0			P_1		P_2			P_3		P_4			P_2		
Visiting page (QA)		A	B		C		D	E		F		G	H		A	D

The user click behaviors are logged in the QA forums (e.g., Yahoo! Answers). An example of user click streams is shown in Table 1.[2] As illustrated in Table 1, five distinct non-QA pages ($P_0 \sim P_4$) and eight QA pages ($A \sim H$) were visited by a user. The click time for these Web pages were recorded. For example, at time stamp 10, QA page A was clicked on from non-QA page P_0. The non-QA page P_0 provided a hyper-link for QA page A, and this user clicked the hyper-link to visit QA page A. Thus, click logs in QA forums record the click behavior of users in detail.

Recommendation of QA pages We develop a framework of QA page recommendation in which, if a user issues a query QA page Q, we recommend a list of QA pages that are relevant to the query Q, where the QA pages in the recommendation list are ranked by their relevance score. Most prior studies recommended QA pages are based on keyword matching methods and those QA pages that contain the issued keyword are ranked by their ratings, as provided by users. Our framework evaluates the relevance degree of QA pages from user click streams without matching keywords or human ratings for QA pages. The overview of our framework is illustrated in Figure 3, where our proposed framework consists of the off-line module and the on-line module. The task in the off-line module is to model a QA latent browsing graph from a given set of click streams. Once the observations of staying time information are collected from each QA page, we utilize BrowseRank on the QA latent browsing graph to derive the importance scores of QA pages. Then, we further derive the relevance degree of the QA pages in the QA latent browsing graph for a given QA page because the QA latent browsing graph contains the direct relevance information between QA pages based on the time-constraint Markov property. In the on-line module, given a QA page, a list of QA pages is derived by exploring Random Walk with Restart in the QA latent browsing graph.

4 Graph structures to model user browsing behavior

Given a set of click streams, we propose two graph structures to capture the user browsing relationships among QA pages. In Section 4.1, the QA browsing graph model is presented. To include more links via latent relationships among QA pages, the *QA latent browsing graph* is developed in Section 4.2.

[2]The snippet of user click streams is from real logs of Yahoo! Answsers after removing privacy information.

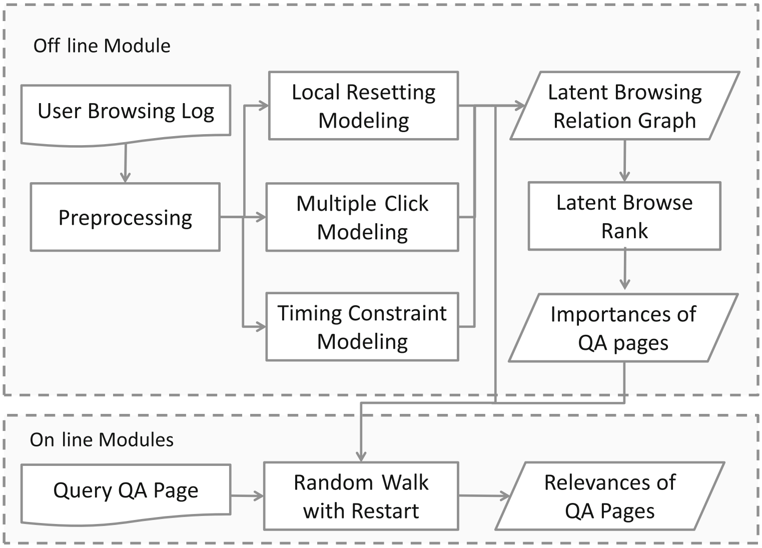


Figure 3 An overview of our proposed QA recommendation.

4.1 QA browsing graph

Since BrowseRank in [11] uses one Browsing Graph to model user browsing behavior, we borrow the concept of Browsing Graph to generate a QA browsing graph. Explicitly, in the QA browsing graph, each node represents one QA page and the edges between QA pages indicate the corresponding browsing relationship. Note that from the user click streams, we define a QA event as follows:

Definition 1 (QA event) A QA event e is a four-tuple: (u, x, y, t), where u is a user ID, y is the QA page, x is the referred page (either non-QA or QA page) that provides a link to QA page y, and t is the time-stamp when QA page y is visited.

A sequence of QA events is obtained by ordering the QA events of a user in an increasing order of time stamps. The users may search for satisfactory answers to a question by visiting QA pages until the QA pages satisfy the need of the user. Consequently, a sequence of chronological QA events that are triggered by a user is derived. Each sequence of QA events from a user represents the relevance of QA pages because the user surfs these QA pages for their questions. Given a sequence of QA events from users, a QA browsing graph is built via the following four steps:

Step 1—session segmentation The sequences of QA events from the collective users were segmented into a set of QA sessions with a given time constraint c because nearby QA events are likely to contain similar QA content. Our segmentation was similar to the time rule of BrowseRank [11]. The definition of a QA session is as follows:

Definition 2 (QA session) Given a timing constraint c, a QA session, s, from a user is a sequence of QA events that are ordered by their time stamp, $s = (e_1, ..., e_r)$, where the time difference between each consecutive QA event is not higher than c.

Given user click streams in Table 1 and the time constraint $c = 60$ s, Table 2 illustrates the result of session segmentation.

Step 2—browsed hyper-link relations We extract the browsed hyper-link relations among the QA pages once the user sessions are determined. A browsed hyper-link relation between QA pages (q_i, q_j) indicate the transition process from QA page q_i to QA page q_j through the hyper-links in q_i. Specifically, the browsed hyper-link transition in the QA pages is defined as follows:

Definition 3 (Browsed hyper-link relations) Given a pair of QA pages (q_i, q_j), a browsed hyper-link relation $r = (q_i, q_j)$ from QA page q_i to QA page q_j occurred when a user reaches q_j through hyper-links in the QA recommendation list of q_i.

The browsed hyper-link relations are explicitly recorded in the QA events. Specifically, given the QA events $e_j = (u_j, x_j, y_j, t_j)$, if the referred page x_j is a QA page, $r = (x_j, y_j)$ demonstrates a browsed hyper-link relation. For example, we observe a browsed hyper-link transition from QA event e_{12} in QA session s_1 in Table 2. We consider each pair of QA pages (q_i, q_j) that are involved in a hyper-link transition as evidence of *context-dependency* between q_i and q_j because they are suggested as relevant by the on-line user who triggered this event. Given a browsed hyper-link transition $r = (q_i, q_j)$, a weighted and directed edge from q_i to q_j is created. Each edge r is associated with a transition frequency, and defined as the number of hyper-link transitions from q_i to q_j from the collective users. For example, in Figure 4, a browsed hyper-link relation $(A \rightarrow B)$ for e_{12} is built with its transition frequency, 1, because the browsing log of one user is available.

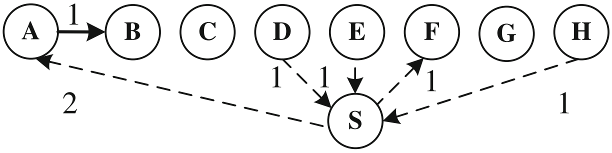
Step 3—global resetting relations The relation among sessions is presented in this step. In BrowseRank, the transition from the end page of a session to the initial page of another session is called a *global resetting relation*. Given two QA sessions

Table 2 Session segmentation.

Session ID	Event	x	y	t
S1	e_{11}	P_0	A	10
	e_{12}	A	B	20
	e_{13}	P_1	C	30
	e_{14}	P_2	D	40
	e_{15}	P_2	E	50
S2	e_{21}	P_3	F	10
	e_{22}	P_4	G	25
	e_{23}	P_4	H	35
S3	e_{31}	P_2	A	110
	e_{32}	P_2	D	120

Figure 4 An example of QA browsing graph.

Relation Type	From	To	Weight
Browsed Hyper-link Relation	A	B	1
Global Resetting Relation	E	S	1
	H	S	1
	D	S	1
	S	A	2
	S	F	1



$s_i = (e_{i1}, ..., e_{im})$ and $s_j = (e_{j1}, ..., e_{jn})$, a global resetting relation indicates a browsing relation from the end QA event of a QA session to the initial QA event in a consecutive QA session. For example, the QA page y_{im} in the QA event e_{im} has a global resetting relation to y_{i1} and y_{j1} in the session s_i and s_j respectively with probabilities in proportion to their frequencies to be an initial QA page.

We follow the technique used in BrowseRank to model global resetting relations. In BrowseRank, two types of QA pages are involved in a global resetting relation, the *end QA pages* and *initial QA pages*. An end QA page refers to the last visited QA page in a QA session. An initial QA page refers to the first QA page in a session. Once the set of end QA pages Q_{end} and the set of initial QA pages Q_{init} are identified, BrowseRank introduces a pseudo vertex, S, to connect the end pages and initial pages. Specifically, a weighted and directed edge $r_{\text{init}} = (S, q_{\text{init}})$ is created for each initial QA page $Q_{\text{init}} \in Q_{\text{init}}$ and a weighted and directed edge $r_{\text{end}} = (q_{\text{end}}, S)$ is created for each end QA page $q_{\text{end}} \in Q_{\text{end}}$. Each edge r_{init} (r_{end}) is associated with a transition frequency, defined as the frequency that q_{init} (q_{end}) initiates (ends) a session. As illustrated in Table 2, the set initial QA pages is $Q_{\text{init}} = \{A, F\}$ and the set of end QA pages is $Q_{\text{end}} = \{E, H, D\}$. Accordingly, we obtained five global resetting relations as illustrated in Figure 4. Among the global resetting relations, $S \to A$ occurred twice and the other global resetting relations occurred once.

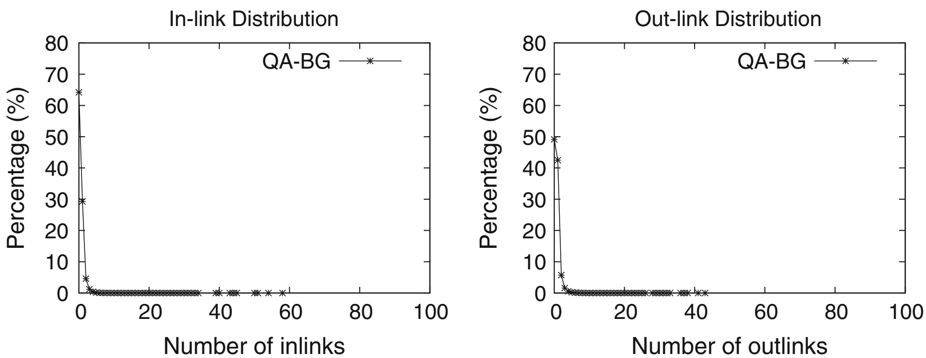
A global resetting transition refers to the behavior in which users drop current sessions and restart from the initial QA pages of available QA sessions. A QA page q_{init} is more likely to be a restart point if q_{init} initiates the QA sessions more frequently. Similarly, y_{end} is more likely to be a drop point if users frequently end sessions after visiting y_{end}. In our example, the pseudo vertex S was used to form a primitive graph (connected graph).

Step 4—staying time extraction For simplicity, given a QA session $s = (e_1, ..., e_r)$ from a user u, the time difference between two consecutive QA events $e_i = (u, x_i, y_i, t_i)$ and $e_{i+1} = (u, x_{i+1}, y_{i+1}, t_{i+1})$ is the staying time for the QA page y_i in e_i. The staying time for the end QA page was randomly determined based on the derived staying times in the corresponding QA session because we could not derive the staying time by subtracting the visiting times of two immediate QA events for the end QA pages.

A *QA browsing graph* $G = (V \bigcup S, E, W)$ is built via the above four steps, where S represents a pseudo vertex, $v \in V$ represents a QA page, and each edge $e \in E$ represents one of the following relations: (i) browsed hyper-link relation; and (ii) the global resetting relations involved in the pseudo vertex. Each QA page $v \in V$ is associated with a set of observations of staying time. The transition matrix of the QA browsing graph is denoted as W, where each entry $w(i, j)$ refers to the transition frequency from v_i to v_j. Given Table 2, the QA browsing graph shown in Figure 4 contains nine vertices, where S represents the pseudo vertex and the remaining vertices indicate the QA pages. As illustrated in Figure 4, the QA browsing graph contains six directed edges, in which one solid edge represent the browsed hyper-link relation and the five dashed edges represent the global resetting relations. The QA browsing graph comprises three isolated QA pages (i.e., C, E and G). The importance of QA pages may be determined by using a QA browsing graph. Given a query QA page, Random Walk with Restart is performed to retrieve the relevant QA pages. The relevant QA pages are derived by traveling the QA browsing graph and their corresponding relevance scores are determined during the traveling the QA browsing graph. However, if most nodes are isolated or have few links, most QA pages may not obtain their importance score and relevance scores. Figure 1 illustrates the degree distribution of QA pages in a QA browsing graph (ignore the direction of edges). As illustrated in Figure 1, approximately 40.5% of QA pages have a zero degree of distribution. Figure 5 illustrates the in-link and out-link distribution of QA pages in a QA browsing graph. As observed in Figure 5, most of the QA pages in the QA browsing graph have zero in-link (out-links). Particularly, 64.2% (49.1%) of QA pages have zero in-links (out-links). Consequently, the link relationships must be enhanced, which requires the design of a latent QA browsing graph.

4.2 QA latent browsing graph

We propose a QA latent browsing graph by improving the links from the latent user browsing behavior. The latent user browsing behavior consists of three relations:



(a) In-link distribution (b) Out-link distribution

Figure 5 Link distributions of QA pages from Yahoo! Asia Knowledge Plus (July 15, 2009–July 17, 2009).

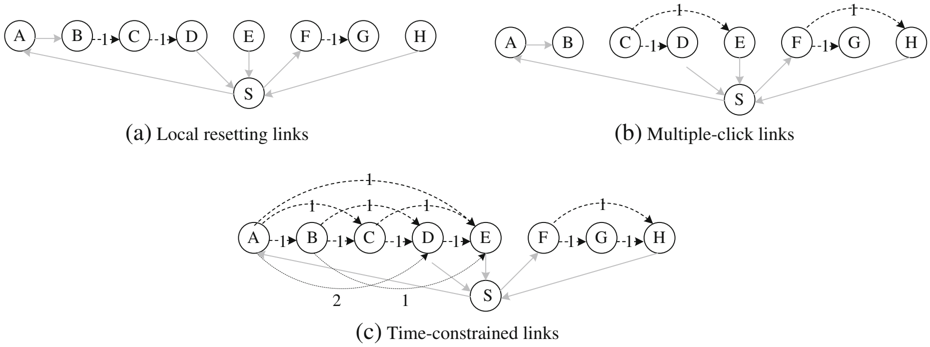
(1) local resetting relations, (2) multiple-click relations, and (3) time-constrained relations. More links are added if QA pages contain these relations. These three relations are presented as follows:

4.2.1 Local resetting relations

In contrast to global resetting relations, which model the transition behavior among QA pages from sessions to sessions, the local resetting relations model the transition behavior for QA pages within a session. We occasionally observe a fragment of a non-QA page sequence between two QA pages. For example, $(e_i, p_1, ..., p_k, e_{i+1})$, where e_i and e_{i+1} represent QA events and p_j represents non-QA events for each $a \leq j \leq k$. An intuitive approach to relate the corresponding QA pages in e_i and e_{i+1} is to construct a path from y_i to y_{i+1} as follows: $y_i \rightarrow \bar{y}_1 \rightarrow ... \rightarrow \bar{y}_k \rightarrow y_{i+1}$. However, users do not always visit the next page through a hyper-link transition from the current page. Instead of configuring the complex relations among the fragment, we simplify their relations for QA pages and connect two corresponding QA pages in e_i and e_{i+1} by generating a direct path from y_i to y_{i+1}. We refer to such relation as *local resetting relation*. The definition of local resetting relation is as follows:

Definition 4 (Local resetting relations) Given a pair of continuous QA events (e_i, e_{i+1}) in a QA session s, a local resetting relation is observed between e_i and e_{i+1} if a continuous sequence of non-QA events between e_i and e_{i+1} is found.

The concept of local resetting relations between two QA pages (y_i, y_{i+1}) is to describe a jump behavior (i.e., from the QA page in e_i, and end at the QA page in e_{i+1} with a minimum of one non-QA page between e_i and e_{i+1}). More links are added among QA pages with local resetting relations. Figure 6a illustrates the examples of local resetting relations (dashed black links). For example, a user may examine the recommended QA pages listed in B, decides to move on to the non-QA page P_1, discover C in P_1, and then decides to visit C. The series of determinations implies that the QA page B is the prior context of C. Furthermore, C is regarded as more relevant or significant than those QA pages that are listed in the recommendation block of B.



(a) Local resetting links

(b) Multiple-click links

(c) Time-constrained links

Figure 6 Examples of QA latent browsing relations.

4.2.2 Multiple-click relations

A multiple-click relation refers to the behavior in which a user visits more than one QA page from a non-QA page. The sequence of QA pages visited through hyper-link transitions from a non-QA page is referred to as a multiple-click group. The current QA page prior to the non-QA page is considered as the context related to each QA page in the multiple-click group. Formally, the multiple-click relation can be defined as follows:

Definition 5 (Multiple-click relations) Given a segment of a QA session $s = (e_i, e_{i+1}.., e_{i+r})$, the QA page y_i in the QA event e_i has a multiple-click relation to each subsequent QA page in e_j, if (1) the referred page x_i in e_i is different from each x_j in the subsequent QA event e_j and (2) each consecutive QA event e_j has the same referred page x_j, where $i + 1 \leq j \leq i + r$.

An example of a multiple-click relation is illustrated by the dashed black links in Figure 6b. A user first examines the QA page C, moves on to the non-QA page P_2 discovers two QA pages $\{D, E\}$ and visits each of them sequentially. The set of QA pages $\{D, E\}$ forms a multiple-click group. The current QA page prior to $\{D, E\}$ is C, which is considered as the context of both D and E. Consequently, there are two multiple-click relations, $C \rightarrow D$ and $C \rightarrow E$.

4.2.3 Time-constrained relations

We explore time-constrained relations for QA pages to improve the QA browsing graph. In contrast to the previously mentioned latent relations, the time-constrained relations may link the QA pages that are not consecutive and improve the neighboring context of each QA page in a session by a given time constraint on this pair. The browsing graph is modeled based on the property of Markov assumption. The Markov assumption assumes that the page that a user will visit next only depends on the current page and is independent of the pages that the user visited previously. The Markov assumption is not realistic because the QA page that a user will visit next may relatively depend on the QA pages that the user visited previously. Table 3 illustrates an example of a QA session. We observed that the first three QA pages were highly relevant to the last QA page event, although the first two QA pages and the last QA page were not repeatedly visited by the user.

To relax the Markov assumption, the time-constrained relation is considered, where an imposed time constraint determines a flexible number of previously visited pages for the QA page that a user will visit next. Specifically, a neighboring time constraint *maxspan* specifies the maximal allowed time difference between a pair of QA pages in a session. The time-constrained relation for a pair of QA pages is defined as follows:

Table 3 An example of time-constrained relation among QA pages in a user QA session.

Time-stamp	Question subject
2009/07/21 21:08:39	About upload in facebook
2009/07/21 21:09:25	How to upload photos to facebook
2009/07/21 21:12:29	How to upload photos to facebook
2009/07/21 21:12:57	Ask for help facebook experts

Definition 6 (Time-constrained relations) Given a QA session $s = (e_1, ..., e_r)$ and a neighboring timing constraint maxspan, a pair of QA pages y_i and y_j has a time-constrained relation if their corresponding QA events e_i and e_j are discontinuously triggered by a user in a session and if the triggered time difference between e_i and e_j is less than maxspan.

A QA browsing graph including time-constrained relations is illustrated in Figure 6c by dashed black links, with the assumption that *maxspan* is 60 s. We observe that the amount of context (previously visited QA pages) that a QA page depends on was extended and restricted by *maxspan*. The *maxspan* affects the fraction of prior context that belongs to a QA page. The QA page that is located in the head of a session has a higher probability to be part of the prior context of the latter QA page in the session. In contrast, the QA page that is located in the end of a session has an optimal prior context. The fraction of prior context is represented by the ratio of incoming and outgoing time-constrained relations. As illustrated in Figure 6c, the QA page that is located in the last position of a session E has the most incoming time-constrained relations, whereas A has the most outgoing time-constrained relations.

Both the local resetting relations and the multiple-click relations that do not satisfy the timing constraint must be eliminated from the QA browsing graph. With the above relations, a QA latent browsing graph is a weighted and directed graph $G^t = (V^t \cup S, E^t, W^t)$, where S represents the pseudo vertex, each vertex $v \in V^t$ represents a QA page, each $e \in E^t$ involved in S represents the global resetting relations, and each $e \in E^t$ connecting two QA pages represent a mixture of the following relations: (i) browsed hyper-link transitions, (ii) local resetting relation (under the time constraint *maxspan*), (iii) multiple-click relation (under the time constraint *maxspan*), and (iv) time-constrained relations. The adjacency matrix of G^t is represented by W^t, where each entry $w^t(i, j)$ denotes the transition frequency from vertex y_i to y_j. Formally, the weight of the edge $e^t(i, j)$ is defined as follows:

$$w^t(i, j) = \begin{cases} \gamma, & \text{if } y_i = S \text{ and } y_j \in V^t \text{ and the QA page } y_j \text{ appears } \gamma \text{ times} \\ & \text{as the first QA event in a QA session} \\ \delta, & \text{if } y_i \in V^t \text{ and } y_j = S \text{ and the QA page } y_i \text{ appears } \delta \text{ times} \\ & \text{as the last QA event in a QA session} \\ c, & \text{if } y_i \in V^t \text{ and } y_j \in V^t \text{ and the transitions from } y_i \text{ to } y_j \\ & \text{ appears } c \text{ times} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

5 Analysis of timing constraint

The timing constraint determines the amount of latent relations and the density of the latent relation browsing graph. The amount of latent relations and the density of the latent relations browsing graph increases in conjunction with *maxspan*. To determine the value of *maxspan*, we address the following issues: (1) identify the elementary features that influence the quality of the relations; (2) examine the effect of these elementary features on the relation qualities as *maxspan* increases; and (3) determine the value of *maxspan*.

5.1 Indicators of the quality of relations

To address the first issue, we examine two aspects of qualities as follows: (i) the relevance degree of a relation, and (ii) the behavioral coherence among a set of relations.

Relevance degree The relevance degree of a relation refers to the relevance of a pair of QA pages is in a relation. If the time difference between the pair of QA pages is short, they are more likely to be relevant to each other. Table 4 presents an example to support our observation in that QA pages involved in short relations tend to be more relevant. Let the QA page on the left hand side of a relation be *prefix* and the QA page on the right hand size be *suffix*. For example, the time difference between QA page A and B is approximately 20 s, and they are considered highly relevant because both pages are related to the topic of "free on-line movies." In contrast, the time difference between QA page A and D is approximately 900 s, and they are considered less relevant because A is about "free on-line movies" and D is about "free on-line music." Furthermore, if a relation, such as (A,B), is supported by several users, the pages are likely to be relevant. Consequently, we present a metric to evaluate the relevance degree of a relation as follows:

$$rel(i, j)^t = \frac{w_{i,j}}{\sum_{k \in N(i)} w_{i,k}}, \tag{2}$$

where $w_{i,j}$ represents the transition frequency of the relation (x_i, x_j) and $N(i)$ represents the neighbors of x_i given a pre-specified maxspan t.

Behavioral coherence The behavioral coherence among a set of relations refers to the similarity of the QA pages that are visited after a particular QA page. Specifically, given a QA page x_i and a maxspan t, if the set of QA pages that are visited after x_i are markedly overlapped, then t may sufficiently identify the relevant subsequent QA pages for x_i. As illustrated in Table 4, we observe that, after u_1 visited QA page A, u_1 sequentially visited B. Similarly, after u_2 visited QA page A, u_2 sequentially visited B and C. Given maxspan as 86 s, we observed the highest number of similar suffix sets for the QA page A because the suffix set of u_1 for A within 86 s was $\{B\}$ and that of u_2 for A within 86 s was $\{B\}$. In contrast, u_1 and u_2 shared less coherence in their browsing behavior when *maxspan* was longer than 86 s. In this case, we discovered that the time difference determined a relative level of behavioral coherence, which

Table 4 Examples of short relations and long relations.

User ID	Prefix	Topic	Suffix	Topic	t (s)
u_1	1509072906787 (A)	Free on-line movies	1008061704665 (B)	Free on-line movies	23
u_2	1509072906787 (A)	Free on-line movies	1008061704665 (B)	Free on-line movies	86
u_2	1509072906787 (A)	Free on-line movies	1008060702361 (C)	Free on-line movies	284
u_3	1509072906787 (A)	Free on-line movies	1508072509613 (D)	On-line mp3 website	919

consequently indicated the relevance degree between the subsequent visited QA pages to a particular QA page.

In summary, the key observations in terms of quality of relations are as follows:

- The relevance degree of a relation increases as the transition frequency increases and the time different decreases.
- The behavioral coherence that is shared among a set of relations may be used to measure the reliability of a timing constraint for a particular QA page.

5.2 Reliability testing

We considere the relevance degree and behavioral coherence to automatically determine the value of the timing constraint, *maxspan*, as follows:

Given a QA page p and a timing constraint t, for each QA event e of visiting the QA page p, we collect those QA pages visited after p within time window t into a suffix set, denoted as sf_e. If the QA page is visited n times, a set of suffix sets of size n, denoted as $Suffix(p) = \{sf_{e1}, ..., sf_{en}\}$, is obtained. We compute a similarity score $sim(sf_{ei}, sf_{ej})$ for each pair of suffix set sf_{ei} and sf_{ej}. The Jaccard coefficient is used to evaluate the similarity of a pair of suffix sets. We then derive an average similarity score from the set of suffix sets. The average similarity score represent the reliability of a timing constraint t for prefix p and may be formally defined as follows:

$$Rp^t = \frac{1}{\binom{|Suffix(p)|}{2}} \sum_{i,j \in Suffix(p), i \neq j} sim\left(sf_{ei}, sf_{ej}\right). \tag{3}$$

We derive a reliability score for each time occurrence when a subsequent QA page was visited after p to determine the optimal timing constraint. Considering the collection of user behaviors in Table 4, four time occurrences are observed after the prefix QA page A (i.e., $t = 23$, $t = 86$, $t = 284$, and $t = 919$). We calculate the reliability score at each time occurrence for the set of corresponding suffix sets and then select the time occurrence with the highest reliability score as the timing constraint t. In this example, $t = 86$ is chosen as the value of *maxspan*. A different prefix QA page may have derived a different timing constraint. This is reasonable because users may behave differently on various QA pages. A number of prefix QA pages are followed by diverse content, which may require a longer *maxspan* for the suffix sets to achieve coherence. On the other hand, a number of prefix QA pages are consistently followed by similar QA pages, which tends to have a shorter *maxspan*. (Table 5).

Table 5 Examples of short relations and long relations.

User ID	Prefix	$t = 23$	$t = 86$	$t = 284$	$t = 919$
u_1	A	B			
u_2	A		B	C	
u_3	A				D
R^t_A		0	0.33	0.167	0.167

6 Algorithms of latent browsing rank and QA recommendation

Given the QA browsing graph structures, the *Continuous-time Markov Model* is used to derive the importance scores for the QA pages in Section 6.1. Then, given a query QA page Q, we adopt *Random Walk with Restart* in the latent browsing graph to recommend relevant QA pages.

6.1 Design of latent browsing rank

We propose *Latent Browsing Rank* (abbreviated as LBR) for computing the importance scores of QA pages. Similar to BrowseRank [11], LBR relies on the continuous-time Markov model. The details of BrowseRank are referred to in [11].

The concept of Browse Rank is to build a model of a continuous-time time-homogeneous Markov process to simulate a random walk on a browsing graph and to use the stationary probability distribution of the process as a measure of page importance. To efficiently estimate the stationary probability distribution of a continuous-time and time-homogeneous Markov process, BrowseRank leverages the correspondence between a continuous-time and time-homogeneous Markov process and a Q-process. Therefore, deriving the stationary probability distribution of a Q-process is a problem in computing the page importance. According to [18], deriving the stationary probability distribution of a Q-process may be reduced to the problem of deriving a stationary probability distribution of the Embedded Markov chain (EMC), which is a discrete-time Markov process. Let the stationary probability distributions of a Q-process be r, where r_i represents the importance of page x_i and the stationary probability distribution of an EMC be $\tilde{r}$. As such, we derive r by the following equation:

$$r_i = \frac{\frac{\tilde{r}_i}{q_{ii}}}{\sum_{j=1}^{n} \frac{\tilde{r}_j}{q_{jj}}}, \tag{4}$$

where q_{ii} represents the parameters in the Q-process and $\tilde{r}_i$ represents the stationary probability of page x_i. Consequently, two tasks are required to determine page importance, as follows: (1) q_{ii} estimation, and (2) deriving the stationary probability distribution $\tilde{r}$ of EMC.

To determine the LBR of QA pages in the QA latent browsing graph (i.e., a QA latent browsing graph $G^t = (V^t \cup S, E^t, W^t)$, where W^t denotes the transition frequency matrix), we perform a column-normalized process to derive M^t for each column in W^t. Then, for each QA page y_i, the parameter q_{ii} that determined the underlying staying time distribution of QA page y_i was determined from a collection of staying time information of y_i by the following equation:

$$\min_{q_{ii}} \left(\left(\mu_i + \frac{1}{q_i} \right) - \frac{1}{2} \left(\sigma_i^2 - \frac{1}{q_i^2} \right) \right)^2, \tag{5}$$

where $q_{ii} < 0$, μ_i represents the average staying time and σ_i represents the variance of staying time.

The ranking vector is defined as $\tilde{r}$ and initialized $\tilde{r}$ with all elements equal to $\frac{1}{n}$, where n is the number of vertices in G^t. Given a damping factor α, which represents the probability of a random surfer remaining in a session instead of resetting to other sessions, the transition probability matrix of the EMC T_{EMC} is estimated as follows:

$$T_{\text{EMC}}(i, j) = \begin{cases} \alpha \frac{w_{i,j}}{\sum_k w_{i,k}} + (1-\alpha)\gamma_j, & \text{if } \sum_k w_{i,k} \neq 0 \\ \gamma_j, & \text{if } \sum_k w_{i,k} = 0 \ , \\ 0, & \text{if } i = j \end{cases} \tag{6}$$

where $w_{i,j}$ denotes the number of transitions from node i to node j and γ_j denotes the global resetting probabilities from the restart node to node j. Once the EMC transition probability matrix T_{EMC} is derived, the stationary probability distribution $\tilde{r}$ is calculated for QA pages by a power iteration algorithm (line 4). The stationary probability distribution $\tilde{r}$ reflects the importance score of QA pages given the underlying relation structure among QA pages. To incorporate the determined staying time distributions into $\tilde{r}$, the stationary probability distribution of the Q-process for QA pages r is computed by (4), where each entry r_i indicated its importance score given the relation structure and staying time distribution (line 5).

Algorithm 1 Latent browsing rank.

Input:
G^t: a QA latent browsing graph,
α: a damping factor.
Output:
r: the vector of importance scores for QA pages.

1 Estimate q_{ii} for each QA page
2 Estimate the transition probability matrix of the EMC T_{EMC}
3 Compute the stationary probability distribution of EMC for QA pages, $\tilde{r}$, by power iteration algorithm
4 Compute the stationary probability distribution of the Q-process for QA pages, r
5 **return** r

6.2 On-line QA recommendation module

The on-line QA recommendation module is presented in this section. Our recommendation module (abbreviated as LBRR) is based on the Latent QA Browsing graph. Specifically, given a query QA page Q, the Latent QA Recommend aims to compute the top-k QA pages that are most relevant to Q. Given a query QA Q, LBRR exploits Random Walks with Restart to retrieve the relevant QA pages in QA browsing graph structures.

Given a QA latent browsing graph G^t, the column-normalized transition probability matrix of G^t is defined as M^t (line 2). The restart vector v_Q and the relevance vector u_Q are initialized with all elements set to zero, except for the entry corresponding to the query QA page Q, which is set to one (line 3–4). Then, the relevance scores of each QA page related to Q is obtained by the Random Walk with Restart model, which iteratively computes the following equation until the ranking vector u_Q converged (line 5–7):

$$u_Q^{k+1} = (1-\alpha) M^t u_Q^k + \alpha v_Q, \tag{7}$$

where k is the number of iterations, u_Q^k represents the relevance score of each QA page in k^{th} iteration, and α is the restart factor, which represents the probability of a random surfer restarting from the current QA page to the query QA page Q instead of following the outgoing relations to its neighboring QA pages in M^t (line 6). Consequently, a QA page y_i in G^t is regarded as more relevant to Q if the probability of a random surfer reaching the QA page y_i from Q is higher. After the relevance vector u_Q^{k+1} is converged, u_Q^{k+1} is returned, where each entry $u_Q(i)^{k+1}$ represents its relevance score to the query QA page Q. Finally, a list of QA pages with top-k highest relevance scores are returned. If the relevance scores of the returned QA pages are the same, their rank may be determined by the LBR score. The algorithm for computing the relevance scores of QA pages for a query QA page Q is summarized in Algorithm 2.

Algorithm 2 Latent browsing recommendation.

Input:
G^t: a QA latent browsing graph
α: a restart probability,
Q : a query QA page.
Output:
u_Q: the vector of relevance scores for query QA page Q.

1 Compute M^t
2 Initialize r_Q
3 Initialize $u_Q = v_Q$
4 **while** u_Q^{k+1} has not converged **do**
5 Update u_Q^k by $u_Q^{k+1} = (1-\alpha)M^t u_Q^k + \alpha v_Q$
6 **end**
7 **return** u_Q^{k+1}: the vector of relevance scores for query QA page Q

7 Performance evaluation

In this section, we first compare the effectiveness of our proposed graph models with the BrowseRank in terms of importance rank of QA pages (Latent Browse Rank) and relevance rank of QA pages (Latent Browse Recommendation Rank). Then, we investigate the robustness of our proposed graph models by varying values of *maxspan*.

7.1 Datasets and experimental settings

We conduct experiments with real datasets to evaluate the performance of the proposed graph model. The dataset is collected over three days during July 2009 from the commercial Question and Answering forums of Yahoo! Asia Knowledge Plus (AKP). The dataset contains approximately 43 millions click events and 5.8 millions clicked QA pages. The dataset is modeled into three types of graphs as follows: browsing graph (BG), QA browsing graph (QA-BG) and QA latent browsing graph

Table 6 Description of graph models.

Graph model	*maxspan* (s)	Number of vertices	Number of edges	Graph density
BG	None	3,375,998	2,708,259	4.75×10^{-7}
QA-BG	None	3,162,988	2,675,099	5.35×10^{-7}
L-QA-BG(t300)	300	5,434,668	27,298,162	1.85×10^{-6}
L-QA-BG(t600)	600	5,496,791	39,235,500	2.60×10^{-6}
L-QA-BG(t1200)	1,200	5,519,518	47,315,599	3.11×10^{-6}

(L-QA-BG). As summarized in Table 6, the browsing graph BG contains both non-QA pages and QA pages; and the QA-BG contains only QA pages. The L-QA-BG only has QA pages with an imposed time threshold *maxspan*, ranging from 300 to 1200 s. Now, we compare the density of graphs, where given a graph, $G = (V, E)$, the graph density is computed as follows:

$$D(G) = \frac{2|E|}{|V|\,(|V|-1)}. \tag{8}$$

7.2 Evaluation metrics

Both the importance ranking quality and relevance ranking quality are measured in terms of three aspects as follows: (i) the amount of incoming transitions, (ii) the accumulated staying time, and (iii) the number of multiple-click groups. Each of these aspects is defined as follows:

Let the ranking list of QA pages generated by Latent BrowseRank on the QA latent browsing graph L-QA-BG(t600) be TOP(L-QA-BG,t600); and the top QA pages generated by BrowseRank on BG and QA-BG be TOP(BG) and TOP(QA-BG), respectively. An indicator of popularity is proposed to measure the distribution of the incoming hyper-link transitions over the rank. Specifically, we define the popularity of a QA page q, IN_q, as the amount of incoming transitions of a QA page q in the QA browsing graph QA-BG. Considering the ranking position, we define the average number of incoming transitions for a QA page q at rank K as follows:

$$\alpha_{q@K} = \frac{\sum_{p=1}^{K} IN_{q@p}}{K}. \tag{9}$$

Accordingly, the popularity measured by mean average incoming transitions of a given ranking list is defined as follows:

$$\beta@K = \frac{\sum_{p=1}^{K} \alpha_{q@K}}{K}. \tag{10}$$

In principle, given a ranking list with popular highly-ranked QA pages, the ranking list achieves a higher score in terms of popularity at a higher rank in comparison with a ranking list with unpopular highly-ranked QA pages.

An indicator of information quality, δ@K, is proposed to measure the distribution of staying time over the rank. Specifically, we define the information quality of a QA page q as the sum of a set of observations of staying time for a QA page q, denoted as γ_q. The longer the total amount of time that users spent on a QA page q, the more informative the QA page is considered. Given a ranking list, the information quality defined by total amount of time that users spent on a top-K ranking list is measured as follows:

$$\delta@K = \sum_{p=1}^{K} \gamma_{q@p}. \tag{11}$$

In principle, given a ranking list with highly-ranked QA pages that are associated with longer accumulated staying time, the ranking list achieves a higher score in terms of information quality at a higher rank in comparison with a ranking list with QA page that are associated with less staying time but are highly-ranked.

The indicator of reference value to evaluate the effectiveness of a ranking list is proposed. Specifically, the value of a QA page q is measured by how the QA page q associates with other QA pages. To measure the reference value of QA pages, we identify multiple-click groups in the collected user behavior dataset and accumulated the occurrences in multiple-click groups for each QA page q. A multiple-click group is a set of QA pages with the same referrer. We maintain a minimum of two QA pages in the dataset for these multiple-click groups. The reference value of a given QA page q is measured by the number of multiple-click groups that q participated in. The more multiple-click groups q participated in, the more QA pages q was associated with. The more QA pages associated with q, the higher the reference value of q. An example is illustrated in Figure 7, in which three multiple-click groups are represented by dotted boxes. The QA page marked by a rectangle is of high reference value because it participates in three multiple-click groups; whereas the QA page marked by a circle is of less reference value because it has no associations. In this case, the QA page represented by rectangle is more valuable than the QA page represented by the circle because the QA page represented by the rectangle frequently interact with other QA pages (Figure 8).

The metric notations with their corresponding descriptions are summarized in Table 7. Let the number of multiple-click groups that a QA page q participates in be η_q. Considering ranking position, the average counts of multiple-click groups of a given a QA page q at rank K is defined as:

$$\epsilon_{q@K} = \frac{\sum_{p=1}^{K} \eta_{q@p}}{K}. \tag{12}$$

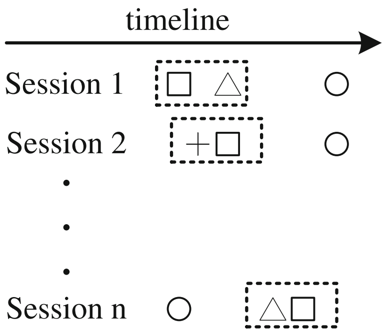


Figure 7 An example of multiple-click groups.

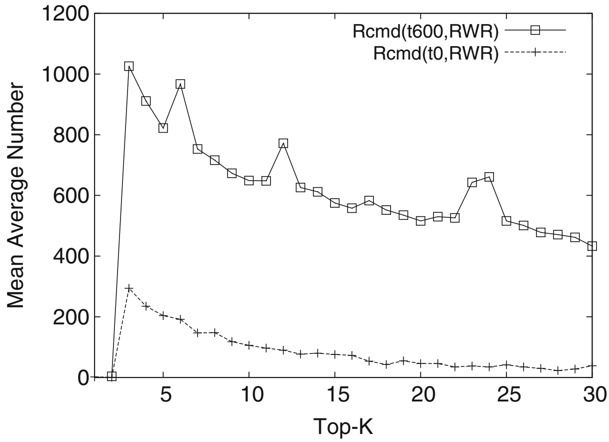


Figure 8 Recommendation quality in terms of mean average number of important QA pages.

Accordingly, given a ranking list of QA pages, the reference value defined by the mean average counts of multiple-click groups at rank K is defined as:

$$\iota@K = \frac{\sum_{p=1}^{K} \epsilon_{q@p}}{K}. \tag{13}$$

Similarly, if the QA pages that more frequently participates in multiple-click groups are ranked higher in a given ranking list, the ranking list achieves a higher score in terms of reference value when K is small.

7.3 Importance ranking quality

In the first set of experiments, we compare the ranking quality in different graph models, as illustrated in Table 6.

7.3.1 Results and discussions

Figure 9 illustrates the comparison of the ranking results in terms of popularity ($\beta@K$), information quality ($\delta@K$) and reference value ($\iota@K$) at varying K. From this figure, we have the following observations.

First, the QA latent browsing graph, L-QA-BG(t600), tends to rank the QA pages of large incoming transitions higher. As Figure 9a demonstrates,

Table 7 Metric notations used in experiments.

Notations	Descriptions
$\alpha_{q@K}$	Average number of incoming transitions for a QA page q at rank K
β@K	Mean average incoming transitions of a given ranking list
γ_q	Sum of a set of observations of staying time for a QA page q
δ@K	Total amount of time that users spent on a top-K ranking list
$\eta_{q@p}$	Number of multiple-click groups that a QA page q participates in
$\epsilon_{q@K}$	Average count of multiple-click groups of a given a QA page q at rank K
ι@K	Mean average count of multiple-click groups at rank K

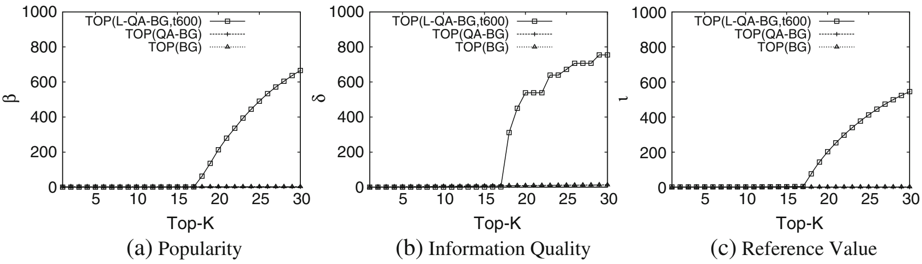


Figure 9 Comparison of ranking quality.

TOP(L-QA-BG,t600) exhibits a significant increase in the popularity score after $K = 18$; whereas the popularity score in TOP(BG) and TOP(BG) is low and stable over K. The QA page ranked at $K = 18$ in TOP(L-QA-BG,t600), denoted as $q@18$, is an effectively-summarized documents that is illustrated with excellent pictures and concise texts. It was visited by over ten thousand users within three days and accumulated a large amount of incoming transitions from other QA pages in the domain-specific browsing graph QA-BG. The QA page ($q@18$) was ranked at 83,642 in TOP(QA-BG). This difference is mainly because BrowseRank is highly sensitive to the number of observations of staying time information. A few observations of staying time information can result in a bias of the underlying staying time of a page. Consequently, those pages with a markedly high standard deviation, but visited by a few people (less than ten), are in TOP(BG). In contrast, popular pages such as $q@18$ have a relatively low average staying time and standard deviation and are regarded as less important by BrowseRank. The popular pages of a higher quality may be highlighted with the help of the transitions that were derived from the implicit information of user-perceived relevance among QA pages. A number of QA pages with high quality in the top-10 of L-QA-BG(t600) are isolated in QA-BG and should not appear in TOP(QA-BG). However, the implicit transitions in the QA latent browsing graph may discover these pages.

Second, *Latent Browsing Rank* may rank those QA pages with excellent information quality as high as illustrated in Figure 9b. Most QA pages in TOP(QA-BG) have a significantly long average staying time or standard deviations, however, they were visited by less than ten people. Unlike BrowseRank, which is sensitive to the staying time distribution, *Latent Browsing Rank* emphasizes the importance of latent relevance transitions. Consequently, the QA pages with a higher accumulated staying time may be ranked higher by *Latent Browsing Rank*.

Third, *Latent Browsing Rank* tends to rank the frequently associated QA pages higher. As Figure 9c demonstrates, the QA pages in TOP(L-QA-BG,t600) participated in more multiple-click groups. The QA pages in TOP(L-QA-BG,t600) are of higher reference value because they are frequently co-visited with other QA pages.

7.3.2 Top-10 QA pages

Table 8 illustrates the top-10 QA pages that were produced by BrowseRank under QA-BG and the top-10 QA pages that were produced by *Latent Browsing Rank* under L-QA-BG(t600). Among the top-1000 QA pages that were produced by QA-BG and L-QA-BG(t600), 0.14% were the same. The first column in Table 8

Table 8 Top 10 QA pages produced by two different graph models.

Rank	TOP (BG)	TOP (L-QA-BG,t600)
1	[*Family*] channels to appeal for sudden surges of water consumption	[*Movie*] the best Chinese Films you have ever seen
2	[*Social and Human*] what's the civil culture?	[*Drama*] TV channel and schedule for nice Taiwanese Cinema
3	[*Investment*] a platform in domestic stock for bidding "stop loss limit order" via intelligent investment tool	[*Movie*] the Websites about the release of Chinese Films
4	[*Mind*] healthy diet for effective defecation	[*Movie*] Taiwanese Films during early stage of Taiwan
5	[*Literature*] a translation for classical Chinese	[*Movie*] where can I buy the Chinese Film "Wolf"
6	[*Social and Human*] comments on navy ship volunteer for military service	[*Movie*] the first Taiwanese films after World War II
7	[*Education*] where can I buy "magic follows"	[*Movie*] TV channel for Chinese Films
8	[*Health Care*] dental clinic near MRT Dingxi Station	[*Movie*] opinions about Taiwanese Films
9	[*Hardware*] related issues in installing driver for Bluetooth devices	[*Movie*] the Websites to search for movies
10	[*Party Politics*] the cause of death of Ching-feng Yin?	[*Movie*] the major Chinese Films in recent ten years

illustrates the rank from 1 to 10. The second and third column summarizes the main idea of the QA page at the corresponding rank.

7.4 Recommendation quality

We select 2000 QA pages of high visiting frequency as the query collection Q to investigate the recommendation quality. For each query $q \in Q$, Random Walk with Restart was performed on QA-BG, BG, and L-QA-BG(t600) to derive the recommendation lists. The recommendation lists that were derived from QA-BG, BG and L-QA-BG(t600) were denoted as Rcmd(QA-BG), Rcmd(BG), and Rcmd(L-QA-BG,t600), respectively. The recommended QA pages for a query q were filtered by a relevance threshold, that is, only the QA pages which are relevant to the query q to certain extent will be recommended. After filtering, Rcmd(BG) contains 1,472 recommendation lists, Rcmd(QA-BG) contains 1,476 recommendation lists and Rcmd(L-QA-BG,t600) contains 1,546 recommendation lists. In addition, a well-known textual relevance model, BM25 [17], is implemented for comparing the recommendation quality. In BG25, the textual content for each QA page is collected and modeled as a set of n-grams. In essence, the title of each QA $q \in Q$ is regarded as the query and the list of QA pages returned by the textual relevance model is regarded as the relevance search result for the QA page q. The collection of relevance search lists of each query $q \in Q$ derived by BM25 are denoted as Rcmd(Text). The recommendation quality of Rcmd(QA-BG), Rcmd(BG), Rcmd(L-QA-BG,t600), and Rcmd(Text) are shown in Figure 10, where the popularity, information quality and reference value of different approaches are presented.

There are several observations from Figure 10. First, the recommendation results derived from all graph models are better than the relevance search results derived

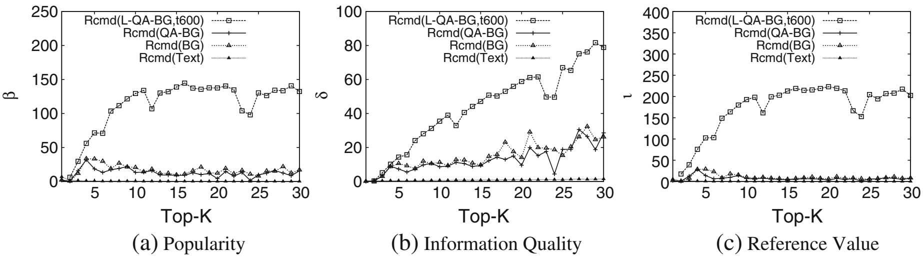


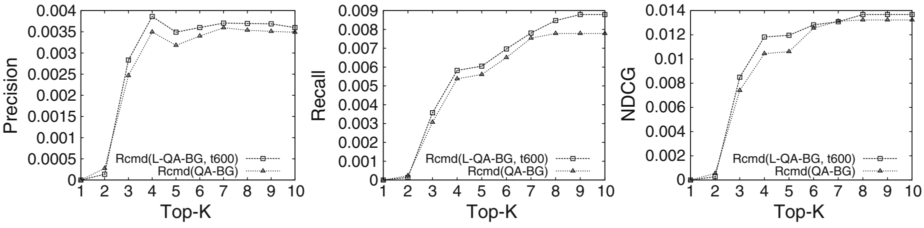
Figure 10 Comparison of recommendation quality.

from the textual relevance model. The main reason is that the keyword-matching technique used in BM25 may fail if QA pages contain the same keyword as in the query QA page, but their contents are totally irrelevant to the topic of the query QA page. On the other hand, BrowseRank and our proposed framework rely on analyzing and modeling user browsing behavior, which avoid the keyword matching problem. Second, Figure 10a indicates that the QA latent browsing graph tends to recommend the QA pages with a high volume of incoming transitions. This is because the QA latent browsing graph emphasizes the factor of linkage information to correct the bias that results from a few sample problems in the staying time distributions. Third, the QA latent browsing graph tends to recommend informative QA pages, in which people are likely to spend a large amount of browsing time. As illustrated in Figure 10b, the accumulated staying time increases dramatically in Rcmd(L-QA-BG,t600) over K. Fourth, the QA latent browsing graph tends to recommend QA pages of higher reference value. As illustrated in Figure 10c, the number of multiple-click groups that were participated by QA pages in Rcmd(L-QA-BG,t600) is markedly higher in comparison with Rcmd(QA-BG) and Rcmd(BG) over K.

From the perspective of page importance, suppose a recommended QA page in Rcmd(L-QA-BG,t600) that is produced by QA latent browsing graph is important if it falls within top-10,000 of TOP(L-QA-BG,t600); and a recommended QA page in Rcmd(QA-BG) is important if it falls within top-100,000 of TOP(BG). As illustrated in Figure 8, at each rank K, Rcmd(L-QA-BG,t600) returns a higher number of important QA pages than Rcmd(QA-BG). Overall, given Q, Rcmd(L-QA-BG,t600) contained 117,196 important QA pages and Rcmd(QA-BG) contained 2,489 important QA pages.

Particularly, we also compare Rcmd(QA-BG) and Rcmd(L-QA-BG,t600) in terms of precision, recall and normalized Discount Cumulative Gain (NDCG) to evaluate the potential effectiveness of the recommendation quality. The dataset is partitioned into training and testing sets according to the time-stamps of records. The amount of records in training data is 80% and the rest 20% of data is regarded as testing data. Suppose the query collection Q is the collection of QA pages associated with each record in testing data. Afterward, for each QA page $q \in Q$, we collect the list of clicked QA pages after q within time interval t as the ground truth GT_q. Let the top-K QA pages returned for the query q be S_q, the precision for the query collection Q is defined as follows:

$$Precision@K = \frac{|GT_q \cap S_q|}{|S_q|}, \forall q \in Q. \tag{14}$$



(a) Precision (b) Recall (c) NDCG

Figure 11 Comparison of recommendation quality in testing set.

The recall for the query collection Q is defined as follows:

$$Recall@K = \frac{|GT_q \cap S_q|}{|GT_q|}, \forall q \in Q. \tag{15}$$

To highlight the ranking quality of recommendation results S_q, we use discounted cumulative gain defined as follows:

$$DCG@K = \sum_{k}^{p=1} \frac{2^{rel(p)} - 1}{\log(1+p)}, \forall q \in Q \tag{16}$$

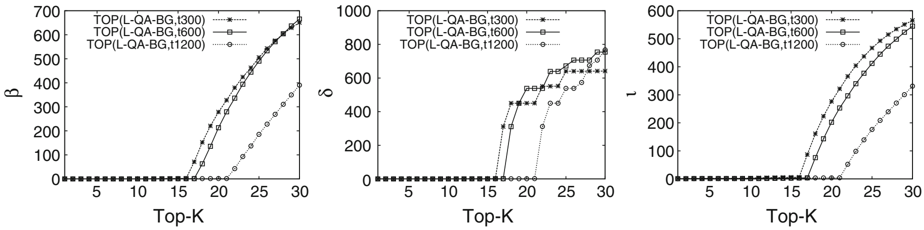
where $rel(p)$=1 if the QA pages ranked at position p falls within GT_q; otherwise, $rel(p)$=0.

Accordingly, the normalized discounted cumulative gain for a query q is computed as:

$$nDCG@K = \frac{DCG@K}{IDCG@K}, \forall q \in Q \tag{17}$$

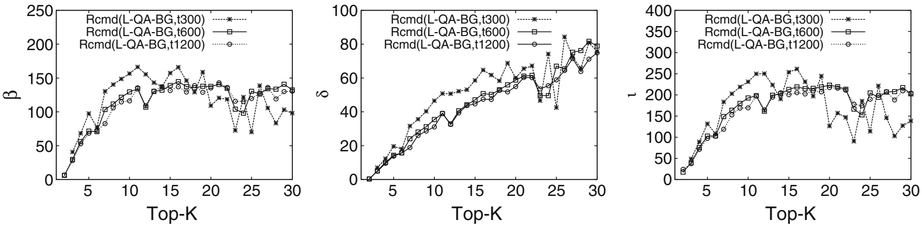
where $IDCG@K$ is the $DCG@K$ of ideal ordering at K.

As shown in Figure 11, the recommendation results derived from QA latent browsing graph have higher precision, recall and nDCG than those derived from QA



(a) Popularity (b) Information Quality (c) Reference Value

Figure 12 Comparison of ranking quality with varying value of *maxspan*.



(a) Popularity (b) Information Quality (c) Reference Value

Figure 13 Comparison of recommendation quality with varying value of *maxspan*.

browsing graph. In other words, QA latent browsing graph potentially guarantees higher on-line user click rates.

7.5 Sensitivity analysis

We illustrate the robustness of the QA latent browsing graph in this section. The effectiveness of importance ranking and recommendation quality is sensitive to the maximal allowed time difference between a pair of QA pages. Table 6 illustrates the details of QA latent browsing graphs with varying values of *maxspan*.

In Figure 12, we observe that the ranking qualities with varying values of *maxspan* displayed similar curves in terms of popularity, information quality, and reference value. A slight difference is that *Latent Browsing Recommendation* with a smaller *maxspan* highly ranks those QA pages with a higher popularity, excellent information quality, and a higher reference value. In Figure 13, the recommendation quality with $maxsapn = 300$ imposed on a QA latent browsing graph notably fluctuate in comparison with $maxsapn = 600$ or higher over K.

8 Conclusions

We developed a framework of QA recommendation in QA forums, such as Yahoo! Asia Knowledge Plus and Microsoft Answers. Explicitly, given a set of user click streams, we proposed a QA browsing graph structure to capture the actual browsing behaviors of the users of QA pages. We further explored the latent browsing relationships among QA pages to improve the links in the QA browsing graph because the QA browsing graph has more isolated QA pages. We also integrated the staying time factor in determining the relevance among all QA pages for a given query QA page. Our experiments on a collection of user browsing logs from Yahoo! Asia Knowledge Plus indicated that the QA latent browsing graph and staying time analysis provides superior performance to those of the baseline ones. By performing Random Walk with Restart in the QA latent browsing graph, our framework recommended QA pages that were highly related to a given query QA page. Thus, due to the QA latent browsing graph, we proposed two types of scores for QA pages (i.e., LBR and LBRR). Thus, our QA recommendation derived a list of QA pages with more information and higher reference value, which indicates that users are willing to spend their time on those pages after reading the query QA pages.

References

1. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding high-quality content in social media. In: Proceedings of the First ACM International Conference on Web Search and Web Data Mining, pp. 183–194. ACM (2008)
2. Bertino, E., Ferrari, E., Perego, A.: A general framework for web content filtering. World Wide Web **13**(3), 215–249 (2010)
3. Bian, J., Liu, Y., Agichtein, E., Zha, H.: Finding the right facts in the crowd: factoid question answering over social media. In: Proceeding of the Seventeenth International Conference on World Wide Web, pp. 467–476. ACM (2008)
4. Bian, J., Liu, Y., Zhou, D., Agichtein, E., Zha, H.: Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In: Proceedings of the Eighteenth International Conference on World Wide Web, pp. 51–60. ACM (2009)
5. Gyongyi, Z., Koutrika, G., Pedersen, J., Garcia-Molina, H.: Questioning Yahoo! Answers. In: Proceeding of the Seventeenth International Conference on World Wide Web (2008)
6. He, J., Li, M., Zhang, H.-J., Tong, H., Zhang, C.: Manifold-ranking based image retrieval. In: Proceedings of the Twelfth International Conference on Multimedia, pp. 9–16. ACM (2004)
7. Jeon, J., Croft, W.B., Lee, J.H., Park, S.: A framework to predict the quality of answers with non-textual features. In: Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 228–235. ACM (2006)
8. Kleinberg, J.: Authoritative sources in a hyperlinked environment. J. ACM (JACM) **46**(5), 604–632 (1999)
9. Langville, A., Meyer, C.: Deeper inside pagerank. Internet Math. **1**(3), 335–380 (2004)
10. Leung, C., Chan, S., Chung, F., Ngai, G.: A probabilistic rating inference framework for mining user preferences from reviews. World Wide Web **14**(1), 1–29 (2011)
11. Liu, Y., Gao, B., Liu, T.-Y., Zhang, Y., Ma, Z., He, S., Li, H.: Browserank: letting web users vote for page importance. In: Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 451–458. ACM (2008)
12. Onuma, K., Tong, H., Faloutsos, C.: Tangent: a novel, 'surprise me', recommendation algorithm. In: Proceedings of the Fifteenth ACM International Conference on Knowledge Discovery and Data Mining, pp. 657–666. ACM (2009)
13. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web (1998)
14. Pak, A., Chung, C.: A wikipedia matching approach to contextual advertising. World Wide Web **13**(3), 251–274 (2010)
15. Pan, J., Yang, H., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In: Proceedings of the tenth ACM International Conference on Knowledge Discovery and Data Mining, pp. 653–658. ACM (2004)
16. Rao, W., Chen, L.: A distributed full-text top-k document dissemination system in distributed hash tables. World Wide Web **14**(5–6), 545–572 (2011)
17. Robertson, S.: Overview of the okapi projects. J. Doc. **53**(1), 3–7 (1997)
18. Stewart, W.: Introduction to the numerical solution of Markov chains. Princeton University Press (1994)
19. Sun, J., Qu, H., Chakrabarti, D., Faloutsos, C.: Neighborhood formation and anomaly detection in bipartite graphs. In: Proceedings of the Fifth IEEE International Conference on Data Mining, pp. 418–425 (2005)
20. Suryanto, M.A., Lim, E.P., Sun, A., Chiang, R.H.L.: Quality-aware collaborative question answering: methods and evaluation. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 142–151. ACM (2009)