Topic-Level Random Walk through Probabilistic Model*

Zi Yang, Jie Tang, Jing Zhang, Juanzi Li, and Bo Gao

Department of Computer Science & Technology, Tsinghua University, China

Abstract. In this paper, we study the problem of topic-level random walk, which concerns the random walk at the topic level. Previously, several related works such as topic sensitive page rank have been conducted. However, topics in these methods were predefined, which makes the methods inapplicable to different domains. In this paper, we propose a four-step approach for topic-level random walk. We employ a probabilistic topic model to automatically extract topics from documents. Then we perform the random walk at the topic level. We also propose an approach to model topics of the query and then combine the random walk ranking score with the relevance score based on the modeling results. Experimental results on a real-world data set show that our proposed approach can significantly outperform the baseline methods of using language model and that of using traditional PageRank.

1 Introduction

Link-based analysis has become one of the most important research topics for Web search. Random walk, a mathematical formalization of a trajectory that consists of taking successive random steps, has been widely used for link analysis. For example, the PageRank algorithm [11] uses the random walk techniques to capture the relative "importance" of Web pages from the link structure between the pages. Intuitively, a Web page with links from other "important" pages is highly possible to be an "important" page. Other random walk based methods have been also proposed, e.g. HITS [7].

Unfortunately, traditional random walk methods only use one single score to measure a page's importance without considering what topics are talked about in the page content. As a result, pages with a highly popular topic may dominate pages of the other topics. For example, a product page may be pointed by many other advertising pages and thus has a high PageRank score. This makes the search system susceptible to retrieve such pages in a top position. An ideal solution might be that the system considers the topics talked about in each page and ranks the pages according to different topics. With such a topic-based ranking score, for queries with different topics (intentions), the system can return different topic-based ranking lists.

* The work is supported by NSFC (60703059), Chinese National Key Foundation Research and Development Plan (2007CB310803), and Chinese Young Faculty Research Funding (20070003093).

Q. Li et al. (Eds.): APWeb/WAIM 2009, LNCS 5446, pp. 162–173, 2009.

Recently, a little effort has been made along with this research line. For example, Topic-sensitive PageRank [5] tries to break this limitation by introducing a vector of scores for each page. Specifically, the method assumes that there are multiple topics associated with each page and uses a bias factor to capture the notion of importance with respect to a particular topic. Nie et al. [10] investigate the topical link analysis problem for Web search and propose Topical PageRank and Topical HITS models. However, these proposed methods have a critical limitation: All topics are predefined, which is not applicable to new domains.

Another problem for Web search is how to "understand" the query submitted by the user. With a query, the user typically wants to know multiple perspectives related to the query. For instance, when a user asks for information about a product, e.g., "iPod touch", she/he does not typically mean to find pages containing these two words. Her/his intention is to find documents describing different features (e.g., price, color, size, and battery) of the product. However, existing methods usually ignore such latent information in the query.

Therefore, several interesting research questions are: (1) How to automatically discover topics from Web pages and how to conduct the random walk at the topic-level? (2) How to discover topics related to a query and how to take advantage of the modeling results for search? To the best of our knowledge, this problem has not been formally addressed, although some related tasks have been studied, such as topical PageRank and query reformulation, which we will further discuss in Section 2.

In this paper, we aim at conducting a thorough investigation for the problem of topic-level random walk. We identify the major tasks of the problem and propose a four-step approach to solve the tasks. The approach can discover topics from documents and calculate a vector of topic-based ranking scores for every page. We apply the approach to academic search. Experiments on a real-world data set show that our method outperforms the baseline methods of using language model and those of using traditional PageRank algorithm.

2 Prior Work

We begin with a brief description of several related work, including: language model [20], LDA [1], random walk [11], and topical PageRank [5] [10].

Without the loss of generality, we represent pages on the Web as a collection of linked documents, i.e. $G = (D, E)$, where D represents all pages (documents), and the directed edge $d_1 \rightarrow d_2 \in E$ suggests the page (document) d_1 has a hyperlink pointing to the page (document) d_2. Table 1 summarizes the notations.

2.1 Language Model

Language model is a classical approach for information retrieval. It interprets the relevance between a document and a query word as a generative probability:

$$P(w|d) = \frac{N_d}{N_d + \lambda} \cdot \frac{tf(w,d)}{N_d} + (1 - \frac{N_d}{N_d + \lambda}) \cdot \frac{tf(w,D)}{N_D} \quad (1)$$

Table 1. Notations

SYMBOL	DESCRIPTION
T	set of topics
D	document collection
E	hyperlinks
V	vocabulary of unique words
N_d	number of word tokens in document d
$L(d)$	out-degree of document d
$r[d, z]$	the ranking score of document d on the topic z
w_{di}	the ith word token in document d
z_{di}	the topic assigned to word token w_{di}
θ_d	multinomial distribution over topics specific to document d
ϕ_z	multinomial distribution over words specific to document z
α, β	Dirichlet priors to multinomial distributions θ and ϕ
γ	preference to the topic-intra transition or the topic-inter transition

where $tf(w, d)$ is the word frequency (i.e., occurring number) of word w in d, N_D is the number of word tokens in the entire collection, and $tf(w, D)$ is the word frequency of word w in the collection D. λ is the Dirichlet smoothing factor and is commonly set according to the average document length in the collection [20]. Further, the probability of the document d generating a query q can be defined as $P_{\mathrm{LM}}(q|d) = \prod_{w \in q} P(w|d)$.

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [1] models documents using a latent topic layer. In LDA, for each document d, a multinomial distribution θ_d over topics is first sampled from a Dirichlet distribution with parameter α. Second, for each word w_{di}, a topic z_{di} is chosen from this topic distribution. Finally, the word w_{di} is generated from a topic-specific multinomial distribution $\phi_{z_{di}}$. Accordingly, the generating probability of word w from document d is:

$$P(w|d, \theta, \phi) = \sum_{z \in T} P(w|z, \phi_z) P(z|d, \theta_d) \tag{2}$$

Wei and Croft have applied the LDA model to information retrieval and obtained improvements [17].

2.3 Random Walk

Considerable research has been conducted on analyzing link structures of the Web, for example PageRank [11] and HITS [7].

Many extensions of the PageRank model have been proposed. For example, Haveliwala [5] introduces a vector of scores for each page, with each score representing the importance of the page with respect to a topic. Nie et al. [10] propose a topical link analysis method for Web search. They first use a classification method to categorize each document and then conduct a category-based PageRank on the network of the documents. However, these methods need to pre-define a set of categories, which is not applicable to a new domain.

Richardson and Domingos [12] propose a method for improving PageRank by considering the query terms in the transition probability of the random walk. However, the method does not consider the topical aspects of documents.

Tang et al. [15] propose a method by integrating topic models into the random walk framework. However, they do not consider the topic-level random walk.

2.4 Query Reformulation

A widely used method for query reformulation is to re-weight the original query vector using user click-through data or pseudo-feedback techniques, e.g. [14].

One type of approach is to expand the query terms with synonyms of words, various morphological forms of words, or spelling errors, e.g. Zhai and Lafferty's model-based feedback [19], Lavrenko and Croft's relevance models [8]. Both approaches use relevant documents to expand queries.

Other approaches consider not solely the input text, but also the behavior of the user or the related community, e.g. the click-through data in a search system log of the user, the entire user group or a specific related users in a community, including discovering semantically similar queries [18]. However, all of these methods do not consider the topical semantics of the query.

3 Our Approach

At a high level, our approach primarily consists of four steps:

1. We employ a statistical topic model to automatically extract topics from the document collection.
2. We propose a topic-level random walk, which propagates the importance score of each document with respect to the extracted topics.
3. We propose a method to discover topics related to the query.
4. Given a query, we calculate the relevance score of a document given the query, based on the discovered topics. We further combine the topic-level random walk ranking score with the relevance score.

3.1 Step 1: Topic Modeling

The purpose of the first step is to use a statistical topic model to discover topics from the document collection. Statistical topic modelings [1] [6] [9] [16] are quite effective for mining topics in a text collection. In this kind of approaches, a document is often assumed to be generated from a mixture of $|T|$ topic models. LDA is a widely used topic model. In this model, The likelihood of a document collection D is defined as:

$$P(\mathbf{z}, \mathbf{w}|\Theta, \Phi) = \prod_{d \in D} \prod_{z \in T} \theta_{dz}^{n_{dz}} \times \prod_{z \in T} \prod_{v \in V} \phi_{zv}^{n_{zv}} \tag{3}$$

where n_{dz} is the number of times that topic z has been used associated with document d and n_{zv} is the number of times that word w_v is generated by topic z.

Intuitively, we assume that there are $|T|$ topics discussed in the document collection D. Each document has a probability $P(z|d)$ to discuss the topic z. The topics with the highest probabilities would suggest a semantic representation for the document d. According to the topic model, each document is generated by

following a stochastic process: first the author would decide what topic z to write according to $P(z|d)$, which is the topic distribution of the document. Then a word w_{di} is sampled from the topic z according to the word distribution of the topic $P(w|z)$.

For inference, the task is to estimate the unknown parameters in the LDA model: (1) the distribution θ of $|D|$ document-topics and the distribution ϕ of $|T|$ topic-words; (2) the corresponding topic z_{di} for each word w_{di} in the document d. We use Gibbs sampling [4] for parameter estimation. Specifically, instead of estimating the model parameters directly, we evaluate the posterior distribution on just z and then use the results to infer θ and ϕ. The posterior probability is defined as: (Details can be referred to [1] [4].)

$$p(z_{di}|\mathbf{z}_{-di}, \mathbf{w}, \alpha, \beta) = \frac{n_{dz_{di}}^{-di} + \alpha}{\sum_z (n_{dz}^{-di} + \alpha)} \cdot \frac{n_{z_{di}w_{di}}^{-di} + \beta}{\sum_v (n_{z_{di}v}^{-di} + \beta)} \tag{4}$$

where the number n^{-di} with the superscript $-di$ denotes a quantity, excluding the current instance (the i-th word token in the query q).

3.2 Step 2: Topic-Level Random Walk

After applying the topic model to the document collection, we obtain a topic distribution for each document. Formally, for each document we use a multinomial distribution of topics $\{p(z|d)\}$ (equally written as θ_{dz}) to represent it. We then define a random walk at the topic level.

Specifically, for every document d, we associate it with a vector of ranking scores $\{r[d, z]\}$, each of which is specific to topic z. Random walk is performed along with the hyperlink between documents within the same topic and across different topics.

For document d_k having a hyperlink pointing to document d_l, we define two types of transition probabilities between documents: topic-intra transition probability and topic-inter transition probability, i.e.,

$$P(d_l|d_k, z_i) = \frac{1}{|L(d_k)|} \tag{5}$$

$$P(d_l, z_j|d_k, z_i) = P(z_j|d_l)P(z_i|d_k) \tag{6}$$

where $P(d_l|d_k, z_i)$ is the transition probability from document d_k to d_l on the same topic z_i; $P(d_l, z_j|d_k, z_i)$ is the transition probability from document d_k on topic z_i to page d_l on topic z_j.

Further we introduce a parameter γ to represent preference to the topic-intra transition or the topic-inter transition. Thus, this transition graph formalizes a random surfer's behavior as follows. The random surfer will have a γ probability to access the same topical content on document d_l and will have a $(1 - \gamma)$ probability to find different topics on d_l.

Given this, similar to PageRank, we can define a general form of the random walk ranking score for each document d as:

$$r[d, z_i] = \lambda \frac{1}{|D|} P(z_i|d) + (1 - \lambda) \sum_{d':d' \to d} \left[\gamma P(d|d', z_i) + (1 - \gamma) \frac{1}{T} \sum_{j \neq i} P(d, z_i|d', z_j) \right] \tag{7}$$

where $P(z|d)$ is the probability of topic z generated by document d; similar to the PageRank algorithm, we also introduce a random jump parameter λ, which allows a surfer to randomly jump to different pages in the network:

3.3 Step 3: Modeling Query

The third step is to find topics related to the query. This is not a necessary step, but it can help find semantic information of the query. Modeling query is not an easy task, as the query is usually short. To ensure the coverage of topics, we perform query expansion, a commonly used method in information retrieval. Specifically, for each word w in the query q, we extract its frequent words in the document collection and add them into the query. We consider words appearing in a window-size of the word w as its co-occurring words, i.e. words before and after the word w. We set the window size as 1. We then apply the topic model on the expanded query and discover the query-related topics. For each word w_{qi} in the expanded query q, we sample its topic z according to the probability:

$$P(z_{qi}|\mathbf{z}_{-qi}, \mathbf{w}_q, \alpha_q, \beta) = \frac{n_{qz_{qi}}^{-qi} + \alpha_q}{\sum_z (n_{qz}^{-qi} + \alpha_q)} \frac{n_{z_{qi}w_{qi}}^{-qi} + n_{z_{qi}w_{di}}^{d} + \beta}{\sum_v (n_{z_{qi}v}^{-qi} + n_{z_{qi}v}^{d} + \beta)} \tag{8}$$

where n_{qz} is the number of times that topic z has been sampled from the multinomial distribution specific to query q; α_q is a Dirichlet prior for the query-specific multinomial; the number n^d with the superscript d denotes that we count the numbers in all documents after inference in Step 1. For example, n_{zw}^{d} denotes the number of times that word w assigned to topic z in all documents.

Essentially, we perform another generative process particularly for the query q. The generative process is analogous to that in Step 1, except that we combine the document modeling results for modeling the query. Specifically, we sample a multinomial θ_q for each query q from a Dirichlet prior α_q; we then, for each word w_{qi} in the query, sample a topic z_{qi} from the multinomial; finally, we generate the word w_{qi} from a multinomial specific to the topic z_{qi}. The generative process accounts for dependencies between query and documents. After the process, we obtain a query-specific topic distribution $\{p(z|q)\}$, which suggests a semantic representation of the query.

3.4 Step 4: Search with Topic-Level Random Walk

The last step is to employ the topic-level random walk for search. Specifically, for each document d we combine the relevance score between the query q and the document, with the ranking score of document d from the topic-level random walk. We calculate the relevance score by

$$P_{\mathrm{LDA}}(q|d, \theta, \phi) = \prod_{w \in q} P(w|d, \theta, \phi) = \prod_{w \in q} \sum_{z \in T} P(w|z, \phi_z) P(z|d, \theta_d) \tag{9}$$

However, the learned topics by the topic model are usually *general* and not *specific* to a given query. Therefore, only using topic model itself is too coarse for search [17]. Our preliminary experiments [15] also show that employing only topic model to information retrieval hurts the retrieval performance. In general, we would like to have a balance between *generality* and *specificity*. Thus, we derive a combination form of the LDA model and the word-based language model:

$$P(q|d) = P_{\mathrm{LM}}(q|d) \times P_{\mathrm{LDA}}(q|d) \tag{10}$$

where $P_{\mathrm{LM}}(q|d)$ is the generating probability of query q from document d by the language model and $p_{\mathrm{LDA}}(q|d)$ is the generating probability by the topic model.

Then, we first consider two ways to combine the relevance score with the random walk ranking score, i.e.,

$$S_{\mathrm{TPR}*}(d,q) = P_{\mathrm{LM}}(q|d) \cdot \left[\prod_{w\in Q}\sum_{z\in T} P(w|z)\cdot P(z|d)\right] \cdot \left[\prod_{w\in Q}\sum_{z\in T} r[d,z]\cdot P(w|z)\right] \tag{11}$$

and

$$S_{\mathrm{TPR}+}(d,q) = \left[(1-t)P_{\mathrm{LM}}(q|d) + t\prod_{w\in Q}\sum_{z\in T} P(w|z)\cdot P(z|d)\right] \cdot \prod_{w\in Q}\sum_{z\in T} r[d,z]\cdot P(w|z) \tag{12}$$

where $P(z|d)$ denotes the probability of document d generating topic z and $r[d,z]$ denotes the importance of the document d on topic z.

The above methods sum up the generative probabilities on all topics. We also consider another combination by taking the query modeling results into consideration. The combination score is defined as:

$$S_{\mathrm{TPRq}}(d,q) = P_{\mathrm{LM}}(q|d) \cdot \sum_{z\in T} r[d,z]\cdot P(z|q) \tag{13}$$

where z_w is the topic selected for the word w in query q in the sampling process (cf. Section 3.3).

3.5 Computational Complexity

We analyze the complexity of the proposed topic models. The topic model has a complexity of $O(I_s \cdot (|D| \cdot \bar{N}_d) \cdot |T|)$, where I_s is the number of sampling iterations and $\bar{N}_d$ is the average number of word tokens in a document. The topic-level random walk has a complexity of $O(I_p \cdot |E| \cdot |T|)$, where I_p is the number of propagating iterations, $|E|$ is the number of hyperlinks in graph G, and $|T|$ is the number of topics.

4 Experimental Results

4.1 Experimental Settings

Data sets. We evaluated the proposed methods in the context of ArnetMiner (`http://arnetminer.org`) [16]. 200 example topics with their representative persons/papers are shown at `http://arnetminer.org/topicBrowser.do`.

We conducted experiments on a sub data set (including 14,134 authors and 10,716 papers) from ArnetMiner. As there is not a standard data set with ground truth and also it is difficult to create such one, we collected a list of the most frequent queries from the log of ArnetMiner. For evaluation, we used the method of pooled relevance judgments [2] together with human judgments. Specifically, for each query, we first pooled the top 30 results from three similar systems (Libra, Rexa, and ArnetMiner). Then, two faculties and five graduate students from

CS provided human judgments. Four-grade scores (3, 2, 1, and 0) were assigned respectively representing definite expertise, expertise, marginal expertise, and no expertise. We will conduct two types of searches (paper search and author search) on this selected collection. The data set was also used in [16] and [15].

Evaluation measures. In all experiments, we conducted evaluation in terms of P@5, P@10, P@20, R-pre, and mean average precision (MAP) [2] [3].

Baseline methods. We use language model (LM), BM25, LDA, PageRank (PR) as the baseline methods. For language model, we used Eq.1 to calculate the relevance between a query term and a paper and similar equations for an author, who is represented by her/his published papers). For LDA, we used Eq.2 to calculate the relevance between a term and a paper/author. In BM25, we used the method in [13] to calculate the relevance of a query and a paper, denoted by $S_{\mathrm{BM25}}(d,q)$. For PageRank $S_{\mathrm{PR}}(d)$, we use equation described in [11]. For Topical PageRank S_{TPR}, we use equation described in [10].

We also consider several forms of combination of the baseline methods, including LM+LDA, LM*LDA, LM*PR, LM+PR, LM*LDA*PR, and BM25*PR.

$$S_{\mathrm{LM+LDA}}(d,q) = (1-t)P_{\mathrm{LM}}(d|q) + tP_{\mathrm{LDA}}(d|q) \tag{14}$$

$$S_{\mathrm{LM*LDA}}(d,q) = P_{\mathrm{LM}}(d|q) \cdot P_{\mathrm{LDA}}(d|q) \tag{15}$$

$$S_{\mathrm{LM*PR}}(d,q) = P_{\mathrm{LM}}(d|q) \cdot \mathrm{PR}(d) \tag{16}$$

$$S_{\mathrm{LM+PR}}(d,q) = (1-t)P_{\mathrm{LM}}(d|q) + t\mathrm{PR}(d) \tag{17}$$

$$S_{\mathrm{LM*LDA*PR}}(d,q) = P_{\mathrm{LM}}(d|q) \cdot P_{\mathrm{LDA}}(d|q) \cdot \mathrm{PR}(d) \tag{18}$$

$$S_{\mathrm{BM25*PR}}(d,q) = S_{\mathrm{BM25}}(d,q) \cdot \mathrm{PR}(d) \tag{19}$$

4.2 Experimental Results

In the experiments, parameters were set as follows: for the LDA model, we set the hyperparameters as $\alpha = 0.1$ and $\beta = 0.1$. The number of topics was set different values (5, 15, and 80). In our topic-level random walk, we set the random jump factor as $\lambda = 0.15$ and ranged the factor γ from 0 to 1.0, with interval 0.1. The combination weight t of LM and LDA (Eq.12) was tested with 0, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0. We tuned the parameters and report the best performance.

Table 2 and Table 3 illustrate the experimental results of our approaches (TPR+, TPR*, and TPRq) and the baseline methods. We see that our proposed methods outperform the baseline methods in terms of most measures. The best performance is achieved by TPR+. We also note that although TPRq combines a query modeling process, it does not perform well as we expected. The problem might be the combination methods used in Eq.13. How to find a better combination is also one of our ongoing work.

4.3 Parameter Tuning

Tuning Parameter γ. Figure 1(a) shows the performance (in terms of MAP) of retrieving papers using TPR+ and TPR* with fixed values of the parameter

Table 2. Performance of retrieving papers

Model	p@5	p@10	p@20	R-pre	MAP	MRR	B-pre
LM	0.4286	0.4000	0.4000	0.1094	0.4876	0.6786	0.2732
LDA	0.0286	0.0429	0.0500	0.0221	0.1272	0.1180	0.0238
LM+LDA	0.4000	0.4714	0.4000	0.1561	0.5325	0.7659	0.2808
LM*LDA	0.4000	0.4714	0.4714	0.1605	0.5009	0.4728	0.3236
TPR	0.7143	0.5143	0.4857	0.2201	0.6311	0.8333	0.3687
BM25	0.4286	0.4571	0.4143	0.1196	0.4712	0.6000	0.2734
LM*PR	0.6571	**0.6714**	0.5214	0.1985	0.6541	0.7976	0.4055
LM+PR	0.4286	0.4286	0.4000	0.2100	0.4918	0.6905	0.3107
BM25*PR	0.6571	0.5429	0.4857	0.2094	0.6115	0.8571	0.3780
LM*LDA*PR	0.6571	0.6000	0.5071	0.2146	0.6404	0.7262	0.4091
TPR+	0.7143	0.5714	0.5286	0.2065	**0.7139**	**1.0000**	0.4193
TPR*	**0.7429**	0.6000	**0.5429**	**0.2255**	0.6961	0.8333	**0.4251**
TPRq	0.6286	0.6286	0.4929	0.1838	0.6235	0.7262	0.3874

Table 3. Performance of retrieving authors

Model	p@5	p@10	p@20	R-pre	MAP	MRR	B-pre
LM	0.4000	0.2714	0.1500	0.3274	0.5744	0.8095	0.3118
LDA	0.0857	0.0571	0.0286	0.0595	0.2183	0.2063	0.0506
LM+LDA	0.6571	0.4429	0.2500	0.5881	0.7347	0.8929	0.6012
LM*LDA	0.5714	0.3714	0.2429	0.5107	0.6593	0.8929	0.4926
TPR	0.3714	0.2286	0.1500	0.2964	0.3991	0.4976	0.2267
BM25	0.4857	0.2857	0.1500	0.4214	0.6486	0.7857	0.3775
LM*PR	0.6857	**0.5286**	**0.2786**	0.6357	0.7864	0.9286	**0.7183**
LM+PR	0.6571	0.4429	0.2500	0.5881	0.7338	0.8929	0.6012
BM25*PR	0.6286	0.4857	0.2714	0.6179	0.7392	**1.0000**	0.6526
LM*LDA*PR	0.6571	0.4857	0.2571	0.6655	0.7661	0.9048	0.6831
TPR+	**0.7143**	0.4857	0.2714	**0.7179**	**0.8029**	**1.0000**	0.7151
TPR*	**0.7143**	0.4857	0.2500	0.6512	0.7424	0.8571	0.6650
TPRq	0.6571	0.4571	0.2714	0.6179	0.7084	0.7857	0.6261

γ (ranging from 0 to 1.0 with interval 0.1), denoted as constant γ. We see that for TPR+, the best performance was obtained at $\gamma = 1$ for retrieving papers. We also tried to set γ as $P(z|d')$, denoted as variable γ. For retrieving authors (as shown in Fig. 1(b)), we tuned the parameter with the setting as that for retrieving papers. We see that the performance of retrieving authors is more stable with different values for γ.

Tuning Parameter $|T|$. For tuning the parameter of topic number $|T|$, we varied $|T|$ with 5, 15, 80. The results for retrieving papers are shown in Fig. 2(a),

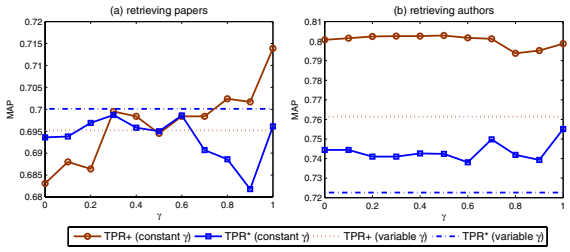

Fig. 1. MAP values of retrieving papers and authors (varying γ)

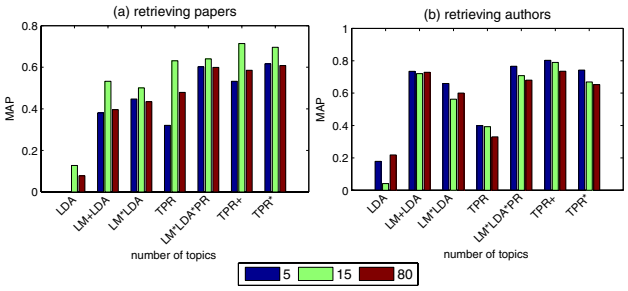

Fig. 2. MAP values of retrieving papers and authors (varying $|T|$)

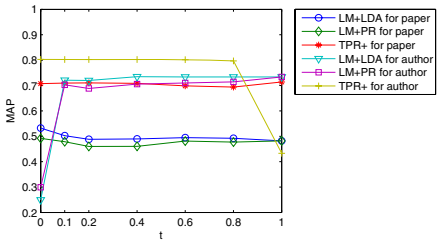

Fig. 3. MAP values of retrieving papers and authors (varying t)

and that for retrieving authors are shown in Fig. 2(b). For papers, we obtained the best performance when setting $|T|$ as 15, for almost all the methods. For authors, the best performance was obtained by setting the topic number as a smaller value 5. It seems that it is more difficult to distinguish the interest of authors than papers.

Tuning Parameter *t*. Shown in Fig. 3, we tuned the parameter t from 0 to 1, with interval 0.2. We see that the performances keep stable when setting t between 0.2 and 0.8.

4.4 Example Analysis

We give a case study to show the motivation of our work. We selected two queries ("natural language processing" and "intelligent agents") to analyze their semantics. For each word in the two queries, we selected the most representative topics for them, i.e. #4, #7, #10, #13 in Table 4. We can see from Table 4 that the query "natural language processing" prefers to topic #10 and "intelligent agents" prefers to topic #4 and #7.

Table 5 shows the importance scores calculated by PageRank and our topic-level random walk (TPR+). When using TPR+ to search "natural language processing", the first document *Verifiable Semantics for Agent Communication Languages* is not retrieved, since the importance score of Topic #10 is small, while the second document *Probabilistic Parsing Using Left Corner Language*

Table 4. Topic distribution for different query words

Query Word	Topic #4	Topic #7	Topic #10	Topic #13
natural	0.000018	0.000018	0.018966	0.000022
language	0.000018	0.002946	0.043322	0.000022
processing	0.000018	0.000018	0.012652	0.000022
intelligent	0.002363	0.022158	0.000023	0.000022
agents	0.037541	0.034784	0.000023	0.000022

Table 5. Importance scores of 4 documents by TPR+ and PageRank

Paper	TPR+				PageRank
	Topic #4	Topic #7	Topic #10	Topic #13	
Verifiable Semantics for Agent Communication Languages	0.000113	0.000026	0.000007	0.000005	0.000612
Probabilistic Parsing Using Left Corner Language Models	0.000002	0.000002	0.000055	0.000014	0.000306
The GRAIL concept modelling language for medical terminology	0.000062	0.000052	0.000050	0.000037	0.003042
Agent-Based Business Process Management	0.000236	0.000179	0.000027	0.000029	0.002279

Models was retrieved. However, when using PagePank to search "natural language processing", the first document was retrieved due to its high score of PageRank, while the second document was not. This result indicates our approach is more reasonable than PageRank. When searching for papers using query "intelligent agents", the fourth document *Agent-Based Business Process Management* can be successfully extracted by our TPR+, while PageRank fails. PageRank chooses the third document *The GRAIL concept modelling language for medical terminology*, which is not retrieved by TPR+, though it has large number of out-links.

5 Conclusion

In this paper we investigate the problem of topic-level random walk. We propose a four-step approach for solving this problem. Specifically, we employ a probabilistic topic model to automatically extract topics from documents. We perform the random walk at the topic level. We also propose an approach to model topics of the query and combine the random walk ranking score with the relevance score based on the modeling results. Experimental results on a real-world data set show that our proposed approach can significantly outperform the baseline methods using language model and those of using traditional PageRank.

There are many potential directions of this work. It would be interesting to investigate how to extend our approach to a heterogeneous network, e.g., a social network consisting of users, documents, and communities. It would also be interesting to investigate how to integrate the user click-through data into the topic model.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
2. Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: SIGIR 2004, pp. 25–32 (2004)
3. Craswell, N., de Vries, A.P., Soboroff, I.: Overview of the trec-2005 enterprise track. In: TREC 2005 Conference Notebook, pp. 199–205 (2005)
4. Griffiths, T.L., Steyvers, M.: Finding scientific topics. In: Proceedings of the National Academy of Sciences, pp. 5228–5235 (2004)
5. Haveliwala, T.H.: Topic-sensitive pagerank. In: Proceedings of the 11th international conference on World Wide Web (WWW 2002), pp. 517–526 (2002)
6. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of SIGIR 1999, pp. 50–57 (1999)
7. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
8. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: Proceedings of 24th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 120–127 (2001)
9. Mei, Q., Cai, D., Zhang, D., Zhai, C.: Topic modeling with network regularization. In: Proceedings of WWW 2008, pp. 101–110 (2008)
10. Nie, L., Davison, B.D., Qi, X.: Topical link analysis for web search. In: SIGIR 2006, pp. 91–98 (2006)
11. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford University (1999)
12. Richardson, M., Domingos, P.: The intelligent surfer: Probabilistic combination of link and content information in pagerank. In: NIPS 2002 (2002)
13. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gatford, M., Payne, A.: Okapi at trec-4. In: Text REtrieval Conference (1996)
14. Rocchio, J.J.: Relevance feedback in information retrieval, pp. 313–323. Prentice Hall, Englewood Cliffs (1971)
15. Tang, J., Jin, R., Zhang, J.: A topic modeling approach and its integration into the random walk framework for academic search. In: ICDM 2008 (2008)
16. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: Extraction and mining of academic social networks. In: KDD 2008, pp. 990–998 (2008)
17. Wei, X., Croft, W.B.: Lda-based document models for ad-hoc retrieval. In: SIGIR 2006, pp. 178–185 (2006)
18. Xue, G.-R., Zeng, H.-J., Chen, Z., Yu, Y., Ma, W.-Y., Xi, W., Fan, W.: Optimizing web search using web click-through data. In: CIKM 2004, pp. 118–126 (2004)
19. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: CIKM 2001, pp. 403–410 (2001)
20. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of SIGIR 2001, pp. 334–342 (2001)