Journal of Web Engineering, Vol. 12, No.3&4 (2013) 265-290

SLASH-BASED RELEVANCE PROPAGATION MODEL FOR TOPIC DISTILLATION

MOHAMMAD AMIN GOLSHANI ALI MOHAMMAD ZAREHBIDOKI VALI DERHAMI

Department of Electrical and Computer Engineering, Yazd University, Yazd, Iran

Golshani.ma@yahoo.com, AliZareh@yazduni.ac.ir, Vderhami @yazduni.ac.ir

Received July 5, 2012
Revised March 1, 2013

An efficient and effective ranking mechanism in the search engines remains as a challenging problem. In recent years, a few relevance propagation models like Hyperlink-based score propagation, Hyperlink-based term propagation, and Popularity-based propagation models have been proposed. In this paper, we will give a comprehensive study of the relevance propagation technologies for Web information retrieval and conduct both theoretical and experimental evaluations over these models to know which model is more effective and efficient. We also propose a new relevance propagation model based on content, link structure (web graph), and number of slashes in the URL. It propagates content and the number of slashes as scores through the link structure. The goal is to find more relevant web pages to the user query. To compare relevance propagation models, Letor 3.0- a standard web test collection- was used in the experiments. We have concluded that using number of slashes in the propagation process provides improvement in Web information retrieval accuracy.

Key words: Web information retrieval, ranking, search engine, propagation methods, number of slashes in the URL
Communicated by: M. Gaedke & P. Fraternali

1 Introduction

Over the past few years, there has been a great deal of research on the use of content and links of Web pages to improve the quality of Web page rankings returned by search engines. When a user sends a query to a search engine, the search engine returns the URLs of documents matching all or one of the terms, depending on both the query operator and the algorithm used by the search engine. Ranking is the process of ordering the returned documents in decreasing order of relevance, that is, so that the "best" answers are on the top. Finding high quality pages is one of the most important challenging issues for any web search engine. To make the web more interesting and productive, we need an effective and efficient ranking algorithm to present more appropriate results to the users. There are thousands or even millions of relevant pages for each query. Nevertheless, users typically consider only the top 10 results. Therefore, we have to focus on the most valuable and appealing web pages. Nowadays, many studies have been done about challenges in Web search engines, such as Web page ranking, web crawling, freshness of the web pages, spam detection, and so on [1, 3, 5, 6, 7, 11, 12, 13, 15, 21, 22, 28, 31, 32, and 33]. In this paper we are going to address the first issue, Web page ranking.

There are currently three major categories of ranking algorithms based on the content and connectivity as the following:

Content based. In traditional IR, the evidence of relevance is thought to reside within the text content of documents. Consequently, the system tries to find documents corresponding to the user query. The fundamental strategy of traditional IR is to rank documents according to their estimated degree of relevance based on measures such as term similarity or term occurrence probability. In order words, for each query the documents with the more similar content to the query will be selected as the more relevant ones. Examples of the content-based ranking algorithms are TF-IDF [23] and BM25 [25].

Connectivity based. In the Web setting, information can reside outside the textual content of documents. For example, links between pages can be used to increase the term based estimation of document relevance. Furthermore hyperlinks, being the most important source of evidence in Web documents, have been the subject of many researches exploring retrieval strategies based on link analysis. Connectivity based algorithms use the links between web pages. They assign a numerical weighting to each element of a hyperlinked set of documents, with the purpose of measuring its relative importance within the set. Instances of the connectivity-based ranking algorithms are PageRank [18] and DistanceRank [33].

Combinational. Using either content-based or connectivity-based algorithms independently, leads to a low-precision ranking function which cannot fully satisfy the users' demands in the Web. Therefore, combination algorithms which use both content and link structure were introduced. In fact they combine content and connectivity information together. These methods can be divided into two groups: one is to enhance link analysis with the assistance of content information, such as HITS and topic-sensetive PageRank [8, 18, 25, 26, and 29] and the other is relevance propagation, which propagate content information with the assistance of the web structure [14, 19, 26, and 27].

In recent years, relevance propagation methods as one of the salient combinational algorithms, has attracted many IR researchers' attention. In the relevance propagation models [14, 19, 26, and 27], the content-based score or query terms are propagated through hyperlinks from one page to another one. In this paper we will give a comprehensive study of relevance propagation technologies (including 12 methods) for Web information retrieval. Then, we will propose a propagation ranking model based on content, connectivity, and number of slashes in the URL. Our idea is based on user browsing behaviour. Baeza-Yates and Castillo [2] have showed pages with less depth (shortest distance in links with the start page(s) of the Web site) are usually more important than pages with more depth. Based on experiment results, we found that the combination of content, link, and URL information in relevance propagation process provides improvement in Web information retrieval accuracy.

The rest of this paper is organized as follows. In section 2, we describe existing relevance propagation models. In section 3, we present the proposed model. In section 4, we study effectiveness of relevance propagation models and report the results of our experiments on the Letor 3.0 -a standard Web test collections. Then we study the efficiency of relevance propagation models in section 5, and finally, in section 6 we give the conclusions and future research directions.

2 Related Work

Ranking has been the subject of extensive research. *Okapi BM25* is a ranking function used by search engines to rank matching documents according to their relevance to a given search query. It is based on the probabilistic retrieval framework developed in the 1970s and 1980s by Stephen E. Robertson, Karen Spärck Jones, and others [24]. BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document. It is not a single function, but actually a whole family of scoring functions, with slightly different components and parameters. One of the most prominent instantiations of the function is as follows.

Given a query Q, containing keywords $q_1,...,q_n$, the BM25 score of a document D is:

$$score(D,Q)=\sum_{i=1}^{n}\frac{N-n(q_i)+0.5}{n(q_i)+0.5}.\frac{f(q_i,D).(k_1+1)}{f(q_i,D)+k_1.(1-b+b.\frac{|D|}{avgdl})} \quad (1)$$

Table 1. BM25 Parameters.

Variable	Definition
$f(q_i,D)$	q_i's term frequency in the document D
$\lvert D\rvert$	Length of the document D in words
$avgdl$	Average document length in the text collection from which documents are drawn
k_1, b	Tuning parameters, $k_1\in[1.2,2.0]$, $b=0.75$
N	The total number of documents in the collection
$n(q_i)$	# of documents containing q_i
$Score(D,Q)$	Similarity between query Q and doc D

Table 2. Relevance propagation models and their abbreviations.

Model			Abbreviation
Score-level	-	Hyperlink based score propagation [26]	*HS*
	Popularity-based	Popularity based Relevance Propagation [14]	*PSH*
Term-level	-	Hyperlink based term propagation model [19]	*HT*
	Popularity-based	Popularity based Relevance Propagation [14]	*PTH*

Many relevance propagation models were developed to propagate content information through the link structure to increase the number of document descriptors. We have grouped them as shown in table 2. Score-level and Term-level methods propagate content similarity between the Web pages and submitted queries as BM25 score and term frequency (TF) through the link structure respectively. For example from table 2, PSH model as a score-level method propagates BM25 score and popularity measure (PageRank score) of the pages, and HT model only propagates TF in the relevance propagation process.

Table 3. Special cases of the relevance score propagation model (HS[a] model).

Special case	Abbreviation	Model formulation
Weighted In-Link	HS-WI	$h^{k+1}(p)=\alpha S(p)+(1-\alpha)\sum_{p_i \to p} h^k(p_i)\omega_I(p_i,p)$ (3)
Weighted Out-Link	HS-WO	$h^{k+1}(p)=\alpha S(p)+(1-\alpha)\sum_{p \to p_j} h^k(p_j)\omega_O(p,p_j)$ (4)
Uniform Out-link	HS-UO	$h^{k+1}(p)=S(p)+(1-\alpha)\sum_{p \to p_j} h^k(p_j)$ (5)

Shakery et al. [26] consider how to use web structure to further improve relevance weighting. They propagated the relevance score of a page to another page through hyperlink between them (web structure). They defined the hyper relevance score of each page as a function of three variables: its content similarity to the query (self-relevance), a weighted sum of the hyper relevance scores of all the pages that point to it (in-link pages), and a weighted sum of the hyper relevance scores of all the pages it points to (out-link pages). According to these definitions, their relevance propagation model can be written as:

$$h^{k+1}(p)=\alpha S(p)+\beta\sum_{p_i\to p} h^k(p_i)\omega_I(p_i,p)+\gamma\sum_{p\to p_j} h^k(p_j)\omega_O(p,p_j) \tag{6}$$

where $\alpha+\beta+\gamma=1$, $h^0(p)=S(p)$, $\omega_I(p_i,p)\propto S(p)$ and $\omega_O(p,p_j)\propto S(p_j)$

$h^k(p)$ is the hyper relevance score of page p after the k-th iteration, $S(p)$ is the content similarity between page p and the query (BM25 score) and ω_I and ω_O are weighting functions for in-link and out-link pages, respectively. For implementation, they have given three special cases of this model: weighted in-link (WI), weighted out-link (WO), and uniform out-link (UO) (Table 3).

QIN et al. [19] proposed another propagation model (a Term-level model), called HT[b] model, it needs to propagate the frequency of query term (TF) in a Web page before adopting relevance weighting algorithms to rank the document. In fact, HT model is an extended version of the HS model and similar to HS, it has three special cases that are shown in table 4.

[a] Hyperlink-based score propagation model

[b] Hyperlink-based term propagation model

Table 4. Special cases of the relevance term propagation model (HT model).

Special case	Abbreviation	Model formulation
Weighted In-Link	HT-WI	$f_t^{k+1}(p)=\alpha f_t^0(p)+(1-\alpha)\sum_{p_i\to p} f_t^k(p_i)\omega_I(p_i,p)$ where , $\omega_I(p_i,p)\propto f_t^0(p)$ (7)
Weighted Out-Link	HT-WO	$f_t^{k+1}(p)=\alpha f_t^0(p)+(1-\alpha)\sum_{p\to p_j} f_t^k(p_j)\omega_O(p,p_j)$ where $\omega_O(p,p_j)\propto f_t^0(p_j)$ (8)
Uniform Out-link	HT-UO	$f_t^0(p)=f_t^0(p)+(1-\alpha)\sum_{p\to p_j} f_t^k(p_j)$ (9)

Table 5. Popularity-based propagation models and their abbreviations.

Model		Abbreviation	Corresponding method
Popularity-based score propagation using hyperlink (PSH model)	Weighted in-link	PSH-WI	HS-WI
	Weighted out-link	PSH-WO	HS-WO
	Uniform out-link	PSH-UO	HS-UO
Popularity-based term propagation using hyperlink (PTH model)	Weighted in-link	PTH-WI	HT-WI
	Weighted out-link	PTH-WO	HT-WO
	Uniform out-link	PTH-UO	HT-UO

Mousakazemi et al. [14] have extended HT and HS models and proposed Popularity-based relevance propagation framework (including PSH and PTH models, table 5). They used the popularity measure of the Web pages (PageRank score) in the propagation process of the relevance propagation methods (table 6). PageRank is a popular ranking algorithm used by Google to measure the importance of the Web pages. PageRank weights each link based on the importance of the document from which it originates and the number of outlinks in the origin document. It models the users' browsing behaviours as a random surfer model [4, 30]. In this model a person surfs the Web by randomly clicking links on the visited pages. When she (PageRank) reaches to a Web page that does not have any outward link, she will randomly jump to another page. PageRank assumes that a user either follows a link from the current page or jumps to a random page on the Web graph. The rank of page j is then computed by the following equation:

$$r(j)=\frac{1-d}{n}+d*\sum_{i\in B(j)} r(i)/o(i) \qquad (10)$$

where *n* is the number of the Web pages, *O(i)* denotes the number of outgoing links from page *i* and *B(j)* shows the set of pages that point to page *j*. Parameter *d*, damping factor, is used to guarantee the convergence of PageRank and remove the effects of sink pages (pages with no outputs).

Table 6. Model formulation of popularity-based relevance propagation models (PSH & PTH models).

Method	**Model formulation**
PSH-WI	$h^{k+1}(p)=\alpha S(p)+(1-\alpha)\sum_{p_i\to p} h^k(p_i)\omega_I(p_i,p)P(p_i)$ $P(p_i)=\frac{-\gamma}{\log(PR(p_i))}$, $\omega_I(p_i,p)\propto S(p)$ (11)
PSH-WO	$h^{k+1}(p)=\alpha S(p)+(1-\alpha)\sum_{p\to p_j} h^k(p_j)\omega_O(p,p_j)P(p_j)$ $P(p_j)=\frac{-\gamma}{\log(PR(p_j))}$, $\omega_O(p,p_j)\propto S(p_j)$ (12)
PSH-UO	$h^{k+1}(p)=S(p)+(1-\alpha)\sum_{p\to p_j} h^k(p_j)P(p_j)$ $P(p_j)=\frac{-\gamma}{\log(PR(p_j))}$ (13)
PTH-WI	$f_t^{k+1}(p)=\alpha f_t^0(p)+(1-\alpha)\sum_{p_i\to p} f_t^k(p_i)\omega_I(p_i,p)P(p_i)$, $P(p_i)=\frac{-\gamma}{\log(PR(p_i))}$, $\omega_I(p_i,p)\propto f_t^0(p)$ (14)
PTH-WO	$f^{k+1}(p)=\alpha f_t^0(p)+(1-\alpha)\sum_{p\to p_j} f_t^k(p_j)\omega_O(p,p_j)P(p_j)$, $P(p_j)=\frac{-\gamma}{\log(PR(p_j))}$, $\omega_O(p,p_j)\propto f_t^0(p_j)$ (15)
PTH-UO	$f_t^{k+1}(p)=f_t^0(p)+(1-\alpha)\sum_{p\to p_j} f_t^k(p_j)P(p_j)$ $P(p_j)=\frac{-\gamma}{\log(PR(p_j))}$ (16)

Above γ is a tuning parameter that was set to 1.4 and *PR(P)* is the PageRank score of page *P*. For simplicity we have listed reviewed models, their structures, and their abbreviations in table 7.

Table 7. Relevance propagation models, their structures and abbreviations.

Model		Score-level	Term-level	Popularity measure	Links		Abbreviation
					Inlink	outlink	
Hyperlink based score propagation [26]	Weighted in-link	√			√		HS-WI
	Weighted out-link	√				√	HS-WO
	Uniform out-link	√				√	HS-UO
Hyperlink based term propagation [19]	Weighted in-link		√		√		HT-WI
	Weighted out-link		√			√	HT-WO
	Uniform out-link		√			√	HT-UO
Popularity-based score propagation using hyperlink [14]	Weighted in-link	√		√	√		PSH-WI
	Weighted out-link	√		√		√	PSH-WO
	Uniform out-link	√		√		√	PSH-UO
Popularity-based term propagation using hyperlink [14]	Weighted in-link		√	√	√		PTH-WI
	Weighted out-link		√	√		√	PTH-WO
	Uniform out-link		√	√		√	PTH-UO

3 Slash-based relevance propagation model

Baeza-Yates and Castillo [2] proposed three probabilistic models for user browsing in “infinite” Web sites. Their models collapse multiple pages at the same level as a single node, as shown in Figure 1. That is, the Web site graph is collapsed to a sequential list. These models aim at predicting how deep users go while exploring Web sites.

3.1 Random surfer models for an infinite Web site

A Web site has considered *S = (Pages, Links)* as a set of pages under the same host name that forms a directed graph. The nodes are *Pages* = $\{P_1, P_2, . . . \}$ and the arcs are Links such that

$(P_i, P_j) \in$ *Links* iff there exists a hyperlink from page P_i to page P_j in the Web site. For random surfing, each page is modelled in Pages as a state in a system, and each hyperlink in Links as a possible transition. This kind of model has been studied by Huberman et al. [9].

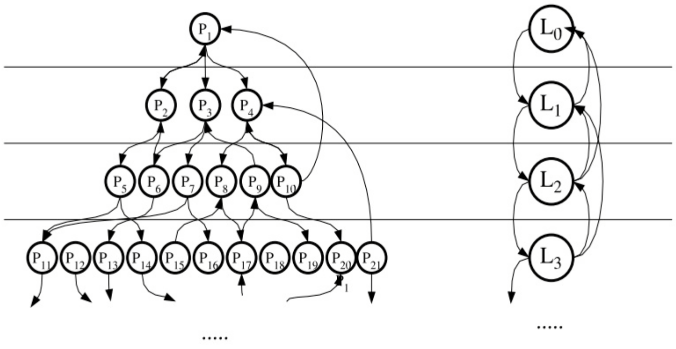

Figure 1. A Web site and a sequence of user actions can be modelled as a tree (left). If we are concerned only with the depth at which users explore the Web site, we can collapse the tree to a linked list of levels (right) [2].

At each step of the walk, the surfer can perform one of the following atomic actions: go to the next level (action next), go back to the previous level (action back), stay in the same level (action stay), go to a different previous level (action prev), go to a different deeper level (action fwd), go to the start page (action start) or jump outside the Web site (action jump).

For action jump an extra node EXIT is added to signal the end of a user session (closing the browser, or going to a different Web site) as shown in Figure 2.

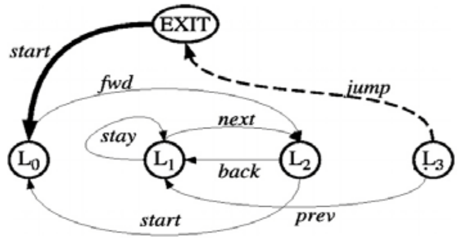

Figure 2. Representation of the different actions of the random surfer. The EXIT node represents leaving the Web site [2].

The set of atomic actions is *A = {next, start/jump, back, stay, prev, fwd}* and the probabilities if the user is currently at level λ, are:

– Pr (next| λ): probability of advancing to the level $\lambda + 1$.

– Pr (back| λ): probability of going back to the level $\lambda - 1$.

– Pr (stay| λ): probability of staying at the same level λ.

– Pr (start, jump| λ): probability of going to the start page of this session, when it is not the previous two cases; this is equivalent in our model to begin a new session.

– Pr (prev| λ): probability of going to a previous level that is neither the start level nor the immediate preceding level.

– Pr (fwd| λ): probability of going to a following level that is not the next level.

In the following three proposed models of random surfing in dynamic Web sites are presented and analyzed.

3.1.1 Model A: back one level at a time

In this model, with probability *q* the user will advance deeper, and with probability *(1 − q)* the user will go back one level, as shown in Figure 3.

Transition probabilities are given by:

– Pr(next| λ) = q

– Pr(back| λ) = 1 − q for $\lambda \geq 1$

– Pr(stay| λ) = 1 − q for $\lambda = 0$

– Pr(start, jump| λ) = 0

– Pr(prev| λ) = Pr(fwd| λ) = 0

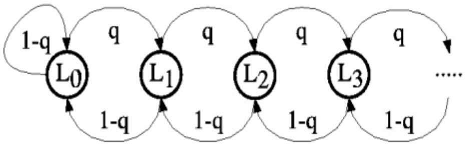

Figure 3: Model A, the user can go forward or backward one level at a time [2].

A stable state *x* is characterized by:

$$x_i = qx_{i-1} + (1-q)x_{i+1} \quad (\forall i \geq 1), \ x_0 = (1-q)x_0 + (1-q)x_1 \tag{17}$$

The solution to this recurrence is:

$$x_i = x_0 (\frac{q}{1-q})^i \quad (\forall i \geq 1) \tag{18}$$

If q ≥ 1/2 then the solution is $x_i = 0$, and $x_\infty = 1$, so we have an asymptotic absorbing state. This means that no depth boundary can ensure a certain proportion of pages visited by the users. When $q < 1/2$ and we impose the normalization constraint, $\sum_{i\geq 0} x_i = 1$, we have a geometric distribution:

$$x_i = (\frac{1-2q}{1-q})(\frac{q}{1-q})^i \tag{19}$$

The cumulative probability of levels *0 . . . k* is:

$$\sum_{i=0}^{k} x_i = 1-(\frac{q}{1-q})^{k+1} \tag{20}$$

This distribution is shown in Figure 4. In this case we have $E\,|\,u\,| = \frac{1-q}{1-2q}$

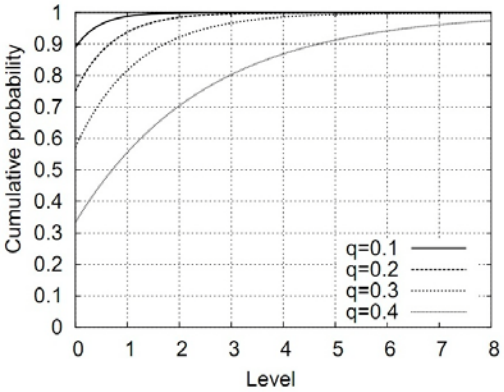

Figure 4. Distribution of visits per depth predicted by model A [2].

3.1.2 Model B: back to the first level

In this model, the user will go back to the start page of the session with probability *(1 − q)*. This is shown in Figure 5.

The transition probabilities are given by:

– Pr (next| λ) = q

– Pr (back| λ) = 1 − q if λ = 1, 0 otherwise

– Pr (stay| λ) = 1 − q for λ = 0

– Pr (start, jump| λ) = 1 − q for $\lambda \geq 2$

– Pr (prev| λ) = Pr(fwd| λ) = 0

A stable state *x* is characterized by:

$$\begin{aligned} x_0 &= (1-q)\sum_{i\geq 0} x_i = (1-q) \\ x_i &= qx_{i-1} \quad (\forall i \geq 1) \end{aligned} \tag{21}$$

and $\sum_{i\geq 0} x_i = 1$.

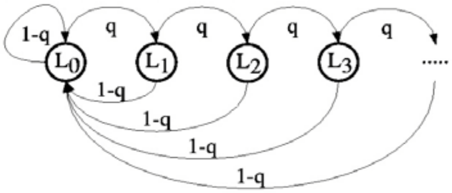

Figure 5. Model B, users can go forward one level at a time, or they can go back to the first level either by going to the start page, or by starting a new session [2].

As we have *q < 1* we have another geometric distribution:

$$x_i = (1-q)q^i \tag{22}$$

The cumulative probability of levels *0…k* is:

$$\sum_{i=0}^{k} x_i = 1 - q^{k+1} \tag{23}$$

This distribution is shown in Figure 6. In this case we have $E(|u|) = \dfrac{1}{1-q}$.

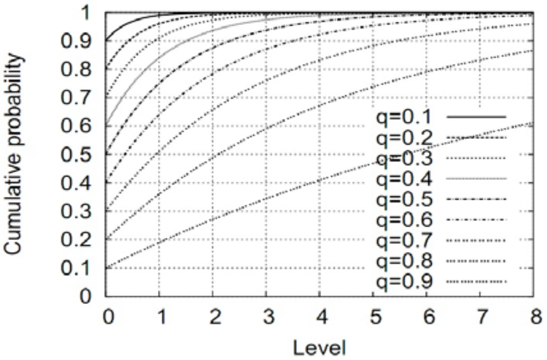

Figure 6. Distribution of visits per depth predicted by model B [2].

3.1.3 Model C: back to any previous level

In this model, the user can either discover a new level with probability *q*, or go back to a previous visited level with probability *(1 − q)*. If a user decides to go back to a previously seen level, the level will chosen uniformly from the set of visited levels (including the current one), as shown in the Figure 7.

– Pr (next|λ) = q
– Pr (back|λ) = 1 − q/(λ + 1) for $\lambda \geq 1$
– Pr (stay|λ) = 1 − q/(λ + 1)
– Pr (start, jump|λ) = 1 − q/(λ + 1) for $\lambda \geq 2$
– Pr (prev|λ) = 1 − q/(λ + 1) for $\lambda \geq 3$
– Pr (fwd|λ) = 0

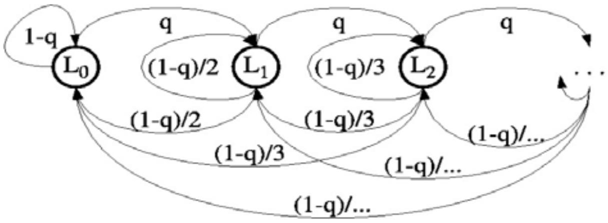

Figure 7. Model C: the user can go forward one level at a time, and can go back to previous levels with uniform probability [2].

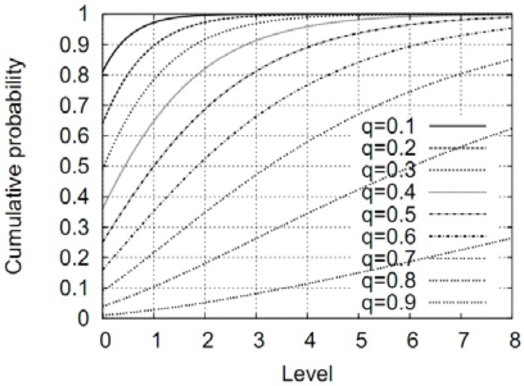

Figure 8. Distribution of visits per depth predicted by model C.

The transition probabilities are given by:

$$x_0 = (1-q)\sum_{k\geq 0}\frac{x_k}{k+1}, \qquad x_i = qx_{i-1} + (1-q)\sum_{k\geq i}\frac{x_k}{k+1} \qquad (\forall i \geq 1) \tag{24}$$

And $\sum_{i\geq 0} x_i = 1$.

We obtain a solution of the form:

$$x_i = x_0(i+1)q^i \tag{25}$$

Imposing the normalization constraint, this yields:

$$x_i = (1-q)^2(i+1)q^i \tag{26}$$

The cumulative probability of levels 0...k is:

$$\sum_{i=0}^{k} x_i = 1-(2+k-(k+1)q)q^{k+1} \tag{27}$$

This distribution is shown in Figure 8. In this case we have $E(|u|) = \dfrac{1}{(1-q)^2}$.

3.1.4 Model comparison

We can see that if *q ≤ 0.4*, depth *3* or *4* captures more than *90%* of the pages a random surfer will actually visit, and if q is larger, say, *0.6*, then depth *6* or *7* captures the same amount of page views.

3.2 The proposed model

According to the obtained results (3.1.4), between two pages relevant to a query, we would tend to favour a page near the top of the directory hierarchy. Usually, pages near the top of the directory hierarchy have less slashes ("/") in their URLs and it seems these pages are more important than others. In this section, we introduce a new ranking model including two propagation methods. We called them *Slash-based Score propagation method* (SS method), and *Slash-based Term propagation method* (ST method).

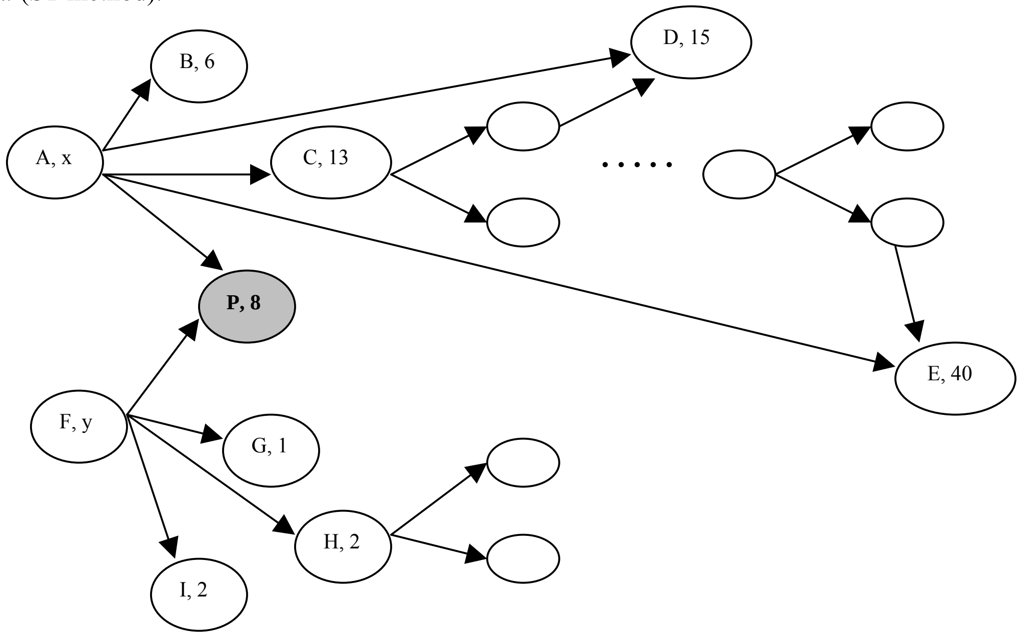

Figure 9. A simple Web graph.

To assign score to the URLs based on the number of slashes that occurs in them, we will introduce a slash weighting function to normalize number of slashes in URLs in range of [0, 1], we called it SWF[c] (p_i, p), $0 \leq SWF(p_i, p) \leq 1$.

$$SWF(p_i, p) = \frac{Slash(URL_p) - \underset{p_k \in O(p_i)}{Min}(Slash(URL_{p_k}))}{\underset{p_k \in O(p_i)}{Max}(Slash(URL_{p_k})) - \underset{p_k \in O(p_i)}{Min}(Slash(URL_{p_k}))} \tag{28}$$

where p is the child page of page p_i , URL_x is Universal Resource Locator (Web page address) of page x, $Slash(URL_x)$ is number of slashes in the URL_x and $O(p_x)$ is a set of the Web pages that have link from p_x. It is clear, the more number of slashes in the URL, the less valuable web page is. According to

[c] Slash Weighting Function

Eq. (27) if there are many slashes in the URL, *SWF(p$_i$, p)* will close to one, otherwise it will close to zero. Therefore, using *(1-SWF(p$_i$, p))* shows URL importance of page *p* in range of *[0, 1].* To show how SWF works, let's see an example in figure 9.

In figure 9, each web page has two symbols, the first one is the page address (URL) and the other is the number of slashes in the URL. For example, *(P, 8)* means the Web page address is *P* and there are *8* slashes in the Web page address. As can be seen, Page *P* has two different parents. Following equations show *SWF* and *(1-SWF)* values of page *P* for each of its parent.

$$SWF(A,P) = \frac{8-6}{40-6} = \frac{1}{17} \approx 0.05,$$

$$1 - SWF(A,P) = 0.95$$

$$SWF(F,P) = \frac{8-1}{8-1} = 1,$$

$$1 - SWF(F,P) = 0.0$$

3.2.1 SS method

In the SS method (the first proposed method), it is assumed that given a page to the user, he reads the content of the page with probability α and he traverses the outgoing edges of its parents with probability $(1-\alpha)$. In Eq. (28), the main formula of iterative SS propagation method is proposed. The SS method propagates BM25 score and number of slashes in URL in the relevance propagation process.

$$h^{k+1}(p) = \alpha s(p) + (1-\alpha)\sum_{p_i \to p}(h^k(p_i) \times (\beta\omega_I(p_i,p) + (1-\beta)\omega_I(p_i,p) \times (1 - SWF(p_i,p)))) \quad (29)$$

where $\omega_I(p_i, p) \propto S(p)$, $O(p_i)$ is out-link of page *i*, $S(p)$ is the content similarity between the page *p* and the query, and *p* is child of *p$_i$*. Because $(1 - SWF(p_i, p))$ can become a big number in compared to ω_I, we use ω_I as a regulator factor to normalize its effect, and β is a balancing parameter to regulate the effects of ω_I and (*1-SWF*) in the propagation process, which according to our experiments can be set to 0.65.

In other words:

Score of the page $p = \alpha$ (Content similarity of page *p*)+ $(1-\alpha) \sum_{i \in \text{p's parents}}$ score of the parent *i* ×

(Balancing Factor × ω_I *+ (1-Balancing Factor)*× ω_I ×*SlashWeightingFunction)*

3.2.2 ST method

This method is similar to the SS method, but instead of BM25 score propagation, it propagates term frequency (TF) between web pages. In other words, it propagates TF and number of slashes in the relevance propagation process. We have used the slash weighting function, $SWF(p_i, p)$, to weight Web page URLs. Eq. (29) shows the main iterative formula of ST propagation method.

$$f_t^{k+1}(p) = \alpha f_t^0(p) + (1-\alpha)\sum_{p_i \to p}(f_t^k(p_i) \times (\beta\omega_{It}(p_i, p) + (1-\beta)\omega_{It}(p_i, p) \times (1 - SWF(p_i, p)))) \quad (30)$$

where $\omega_I(p_i, p) \propto f_t^0(p), O(p_i)$ is out-link of page i, $f_t^k(p)$ is the term frequency of term *t* in page *p* in k-th iteration, and *p* is child of p_i, and β is a balancing parameter, which according to our experiments is set to 0.5.

4 Empirical Evaluations

In this section we are going to evaluate the performance and effectiveness of the proposed model against the old ones. Firstly, we investigate experimental settings, some implementation issues and the evaluation measures and then the results of the effectiveness evaluation are shown.

4.1 Experimental Settings

For the purpose of "Effectiveness Evaluation", we used the ".GOV" corpus of the LETOR 3.0 [20]. LETOR is a benchmark collection for the research on learning to rank for IR, released by Microsoft Research Asia (MSRA). LETOR 3.0 contains standard features, relevance judgments, data partitioning, evaluation tools, and several baselines, for the OHSUMED and the .GOV data collection. Version 3.0 was released in December, 2008. The .GOV corpus, which is crawled from the .gov domain in January, 2002, has been used as the data collection of Web Track since TREC 2002. There are totally 1,053,110 pages with 11,164,829 hyperlinks in it. As our query set, we used the topic distillation task in Web Track 2003 and 2004 (with 50 and 75 queries, respectively). Topic distillation aims to find a list of entry points of good websites principally devoted to the topic. The focus is to return entry pages of good websites rather than the web pages containing relevant information, because entry pages provide a better overview of the websites.

4.2 Constructing the Working Set

Following other researchers [14, 19, 26, and 27], instead of running our experiments on the whole set of data, for each query, we construct a working set. To construct the working set, we first find the top *400* pages with the highest BM25 score as the core set. Then, we expand the core set to the working set by adding the set of pages that point to the core set (citing set) and the set of the pages that are pointed by the core set (cited set). The citing and cited sets are among relevance set pages. Both in construction of the working set and ranking the documents after propagation in term-level methods, we used BM25 as the relevance weighting function. The flowchart of the working set construction is shown in figure 10.

$$WorkingSet = (CoreSet \cup CitingSet \cup CitedSet) \\ \cap \mathrm{Re}\, levantSet$$

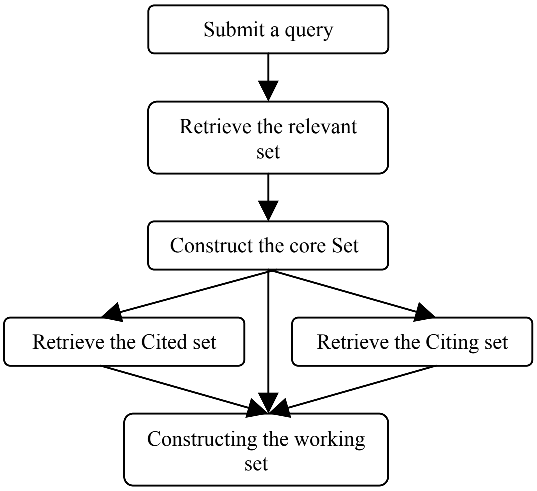

Figure 10. The flowchart of the working set construction.

We do an additional step in the working set construction of the ST method. Usually the TFs of the search query terms in the core set pages compared to other pages in the working set are big numbers (it is common a term has been occurred a lot in a page to show that the page is relevant to a popular topic, in fact it is a form of spamming), so before propagation we reduce term frequency of the search query terms in the core set pages by factor of $\frac{0.8}{Slash(URL_{p\in CoreSet})}$, (e.g. $\frac{0.8}{Slash(URL_{p\in CoreSet})} \times TF_{term_i\ In\ Page\ P}$), in another words, more depth (number of slashes in the URL), more reduction of the TFs will be happen in the Web pages of the core set. This small change leads to significant improvement in the results.

4.3 Evaluation Measures

For the purpose of evaluation, we use a number of evaluation measures commonly used in information retrieval, namely Precision at n (P@n) [20], Mean Average Precision (MAP) [20], and Normalized Discount Cumulative Gain (NDCG) [10].

4.3.1 Precision at n (P@n)

As it has been quoted in reference [20], precision at n measures the relevance of the top *n* documents in the ranking list with respect to a given query:

$$p@n = \frac{\#\text{of relevance docs in top n results}}{n} \tag{31}$$

4.3.2 Mean Average Precision (MAP)

The average precision *(AP)* [20] of a given query is calculated as Eq. (32), and corresponds to the average of $p@n$ values for all relevant documents:

$$AP = \frac{\sum_{i=1}^{N} (\text{P@i} * \text{rel(i)})}{\#\text{total relevant docs for this query}} \tag{32}$$

where *N* is the number of retrieved documents, and *rel(n)* is a binary function that evaluates to *1* if the *n-th* document is relevant, and *0* otherwise. Finally, *MAP* is obtained by averaging the *AP* values over the set of queries.

4.3.3 Normalized Discount Cumulative Gain (NDCG)

For a single query, the NDCG value of its ranking list at position n is computed by Eq. (33):

$$NDCG(n) = Z_n \sum_{j=1}^{n} \begin{cases} 2^{r_j} - 1, j = 1 \\ \dfrac{2^{r_j} - 1}{\log(j)}, j > 1 \end{cases} \tag{33}$$

where *r(j)* is the rating of the *j-th* document in the ranking list, and the normalization constant Z_n is chosen so that the perfect list gets NDCG score of 1. For Letor 3.0, there are two ratings {0, 1} find corresponding to "relevant" and "not relevant" in order to compute NDCG scores.

4. 4 Effectiveness Evaluation

In this section, we present an experimental evaluation of the proposed model against the corresponding models (old ones). For ease of reference, the abbreviations of the proposed model and its corresponding models are shown in table 8.

Table 8. Slash-based propagation methods, theirs abbreviations and corresponding models.

Model		Abbreviation	Corresponding methods
Slash-based relevance propagation	Slash-based Score propagation method	SS	Baseline (BM25) HS (WI, WO, UO) PSH (WI, WO, UO)
	Slash-based Term propagation method	ST	Baseline (TF) HT (WI, WO, UO) PTH (WI, WO, UO)

Following figures and tables compare algorithms in terms of *MAP, p@n, and NDCG@n.*

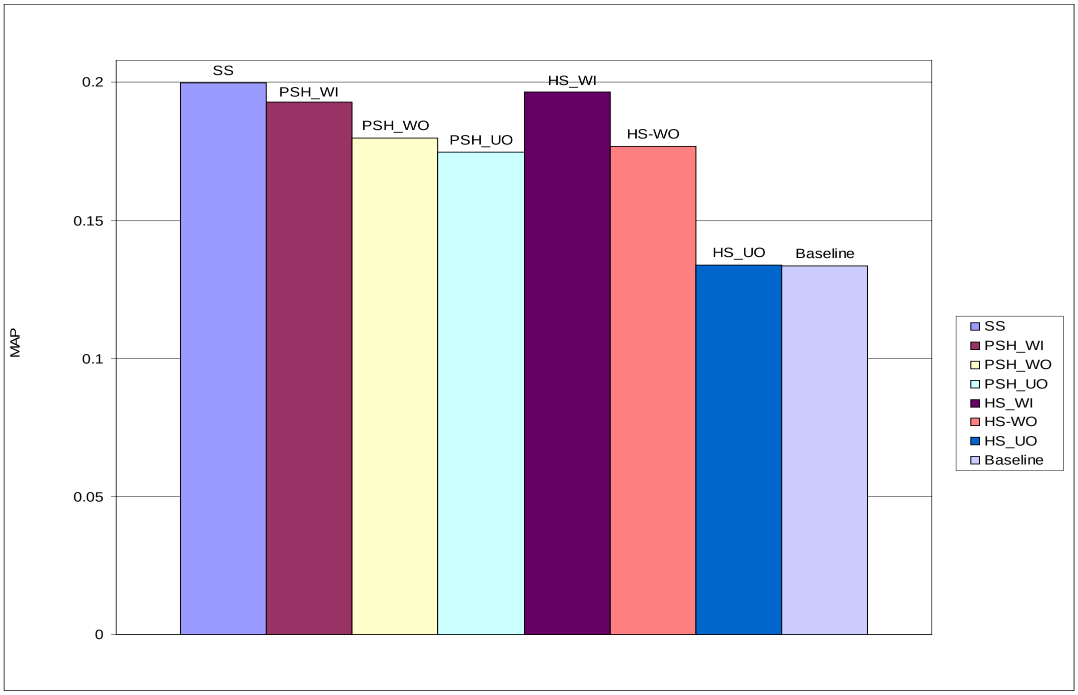

Figure 11. Evaluation of score-level models on the best MAP in TREC 2003.

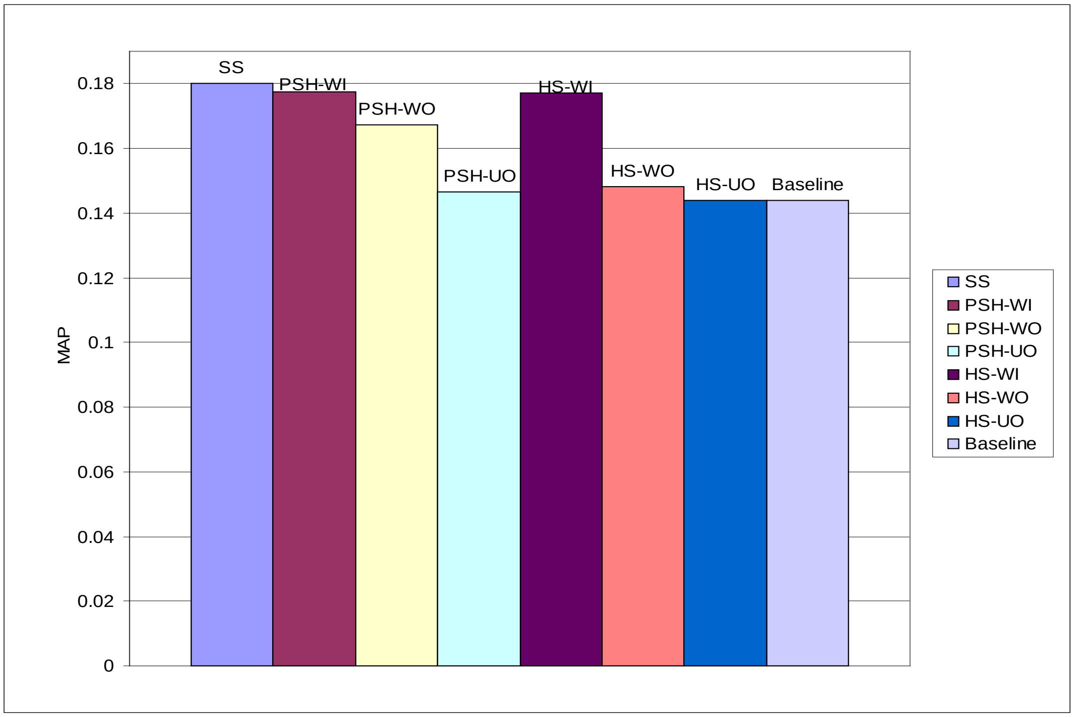

Figure 12. Evaluation of score-level models on the best MAP in TREC 2004.

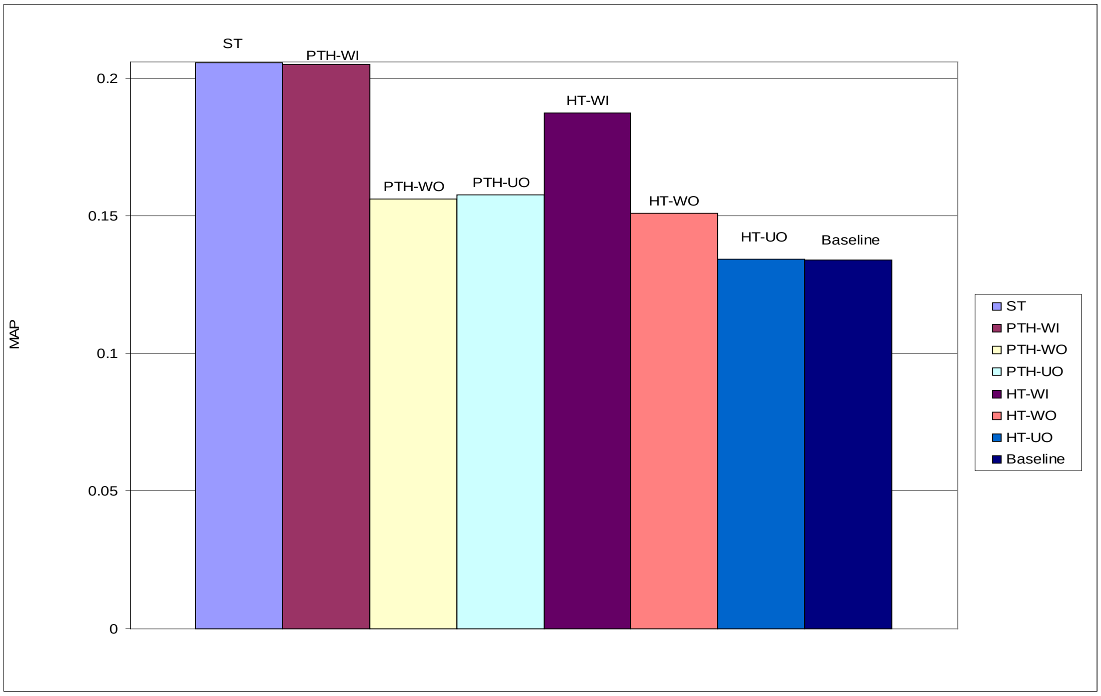

Figure 13. Evaluation of term-level models on the best MAP in TREC 2003.

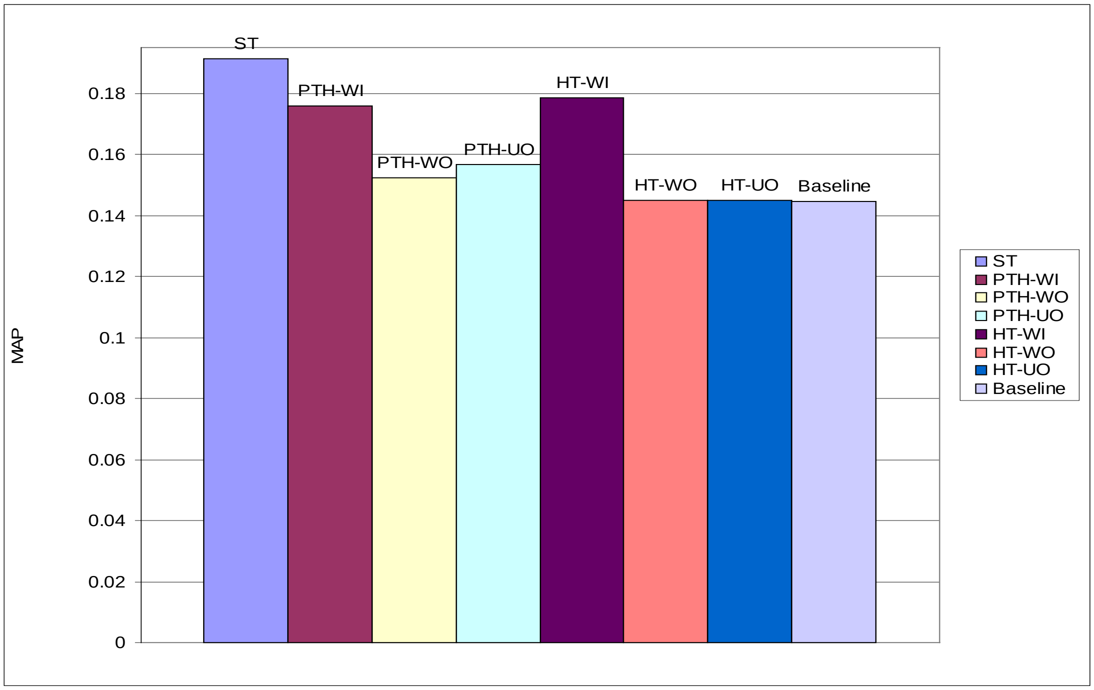

Figure 14. Evaluation of term-level models on the best MAP in TREC 2004.

Table 9. Performance comparison of score-level methods (In terms of MAP)

Method	**Improvement of SS compared to :**
Baseline	37%
PSH-WI	3%
PSH-WO	10%
PSH-UO	18.5%
HS-WI	2%
HS-WO	27%
HS-UO	38%

Table 10. Performance comparison of term-level methods (In terms of MAP)

Method	**Improvement of ST compared to :**
Baseline	42%
PTH-WI	5%
PTH-WO	29%
PTH-UO	26%
HT-WI	9%
HT-WO	34%
HT-UO	43%

Figures 11-14 show the performance of the propagation models (in terms of MAP) on TREC-2003 and TREC-2004 data sets. As can be seen ST and SS are the best methods in term-level and score-level models, respectively. From these figures we can find that the ST method was the most robust, which won the others with most values of MAP in both of the TREC-2003 and TREC-2004 data sets. After ST, SS is the second winner. Besides the peak of the performance value, the robustness of an algorithm is also an important factor for its effectiveness. According to the figures we can see that the PTH-WI method [14] does not have this feature, and different behaviour is shown by PTH-WI in TREC-2003 and TREC-2004 data sets. In other words the extended version of the algorithm (PTH-WI) has worse performance compared to the orginal version of the algorithm (HT-WI) in the TREC-2004 data set. Moreover, Mousakazemi et al. [14] has claimed that HS-WI for $\alpha = 0.85$ has the best performance, but our experiments and QIN et al. [19] show HS-WI for $\alpha = 0.97$ has the best performance and PSH-WI (extended version of HS-WI) does not have any improvement compared to HS-WI.

According to the experiments, we found that the slash-based propagation model, ST and SS methods, can generally outperform their corresponding models. Tables 9 and 10 show improvement of the proposed methods compared to the corresponding methods.

From table 10, as can be seen ST outperforms their corresponding methods (Baseline, PTH-WI, PTH-WO, PTH-UO, HT-WI, HT-WO, HT-UO) by 42%, 5%, 29%, 26%, 9%, 34%, and 43%, respectively. Following tables provide more details.

Table 11. The best performance of each algorithm (in terms of p@n).

Algorithm		α	**GOV with TD-2003**					**GOV with TD-2004**				
			MAP	**P@1**	**P@2**	**P@3**	**P@10**	**MAP**	**P@1**	**P@2**	**P@3**	**P@10**
score-level	Baseline	-	0.1335	0.14	0.12	0.14	0.098	0.1443	0.240	0.2266	0.2089	0.146
	SS	**0.97**	**0.200**	**0.32**	**0.28**	**0.2333**	**0.126**	**0.1802**	**0.28**	**0.2933**	**0.2711**	**0.2067**
	PSH-WI	0.80	0.1927	0.28	0.26	0.2267	0.126	0.1773	0.28	0.2733	0.2711	0.2053
	PSH-WO	0.4	0.1797	0.26	0.22	0.16	0.122	0.1671	0.33	0.2867	0.2756	0.1893
	PSH-UO	0.9	0.1746	0.20	0.21	0.1667	0.118	0.1466	0.2267	0.22	0.2178	0.1547
	HS-WI	0.97	0.1963	0.30	0.26	0.2333	0.128	0.1772	0.28	0.28	0.2578	0.2053
	HS-WO	0.9	0.1520	0.18	0.18	0.1533	0.11	0.1481	0.2667	0.2067	0.1956	0.1627
	HS-UO	1	0.1337	0.14	0.12	0.14	0.098	0.144	0.24	0.2267	0.2089	0.1467
term-level	Baseline	-	0.134	0.14	0.12	0.1333	0.94	0.144	0.253	0.226	0.204	0.146
	ST	**0.7**	**0.2055**	**0.34**	**0.29**	**0.2267**	**0.13**	**0.1912**	**0.3867**	**0.34**	**0.2933**	**0.2093**
	PTH-WI	0.1	0.2049	0.28	0.23	0.2333	0.122	0.1757	0.34	0.3133	0.2889	0.1893
	PTH-WO	0.3	0.1561	0.12	0.14	0.133	0.106	0.1521	0.2	0.24	0.2311	0.1507
	PTH-UO	0.8	0.1577	0.18	0.16	0.1668	0.11	0.1565	0.2667	0.2467	0.2356	0.1627
	HT-WI	0.8	0.1875	0.26	0.25	0.2267	0.124	0.1785	0.3333	0.26	0.2667	0.192
	HT-WO	0.85	0.1510	0.1	0.13	0.1333	0.112	0.145	0.2533	0.1867	0.1867	0.1467
	HT-UO	1	0.1341	0.14	0.12	0.1333	0.096	0.1447	0.2533	0.2267	0.2044	0.1467

For a fair comparison, we have used average value of α for the algorithms that are not robust enough, such as PSH-WI. In other words, if an algorithm for two values of α has the best performance in TREC-2003 and TREC-2004 data sets, average of α is considered in the experiments. In general, methods based on out-link (WO & UO) have lower performances compared to in-link (WI) based methods. Table 12 depicts NDCG@10 of each algorithm over best MAP on the both data sets. From these tables, we can find that the slash-based methods can generally outperform their corresponding methods.

Table 12. The best performance of each algorithm (in terms of NDCG@10).

Algorithm		GOV with TD-2003		GOV with TD-2004	
		α	NDCG @10	α	NDCG @10
score-level	Baseline	-	0.1796	-	0.1872
	SS	**0.97**	**0.2629**	**0.97**	**0.2657**
	PSH-WI	0.80	0.2550	0.80	0.2608
	PSH-WO	0.4	0.2345	0.4	0.2433
	PSH-UO	0.9	0.2298	0.9	0.2016
	HS-WI	0.97	0.2599	0.97	0.2595
	HS-WO	0.9	0.2063	0.9	0.1979
	HS-UO	1	0.1798	1	0.1898
term-level	Baseline	-	0.177	-	0.189
	ST	**0.7**	**0.2750**	**0.7**	**0.2801**
	PTH-WI	0.1	0.2515	0.1	0.2524
	PTH-WO	0.3	0.1973	0.3	0.2026
	PTH-UO	0.8	0.2081	0.8	0.2145
	HT-WI	0.8	0.2518	0.8	0.2475
	HT-WO	0.85	0.2033	0.85	0.1881
	HT-UO	1	0.1777	1	0.1885

To summarize the above experiments, we can draw the following conclusion:

1. Slash-based propagation model was more successful than hyperlink and popularity-based propagation models.
2. Unlike the hyperlink-based and slash-based propagation models, popularity-based propagation models [14] have the overhead of the popularity measure computation. The offline complexity of PageRank equals to $O(100*|E|)$ where $|E|$ denotes the number of edges in the web graph.
3. The proposed relevance propagation model has two methods, ST and SS which the former is more effective than the latter.
4. The SS and ST methods have the best performance for $\alpha = 0.97$ and $\alpha = 0.7$ in TREC-2003 and TREC-2004 data sets, respectively.
5. WI-based methods have better performance compare to WO and UO-based methods.

5 Efficiency Evaluation

In the previous section, we investigate the effectiveness of the relevance propagation models. However, for real-world applications, efficiency is another important factor besides effectiveness [19]. In this regard, we evaluate the efficiency of the models in this section to see their potential of being used in search engines.

Roughly speaking, typical architecture of a search engine has three components [2, 27]: crawler, indexer, and searcher. If we want to integrate relevance propagation technologies into search engine, we should consider these three components. Clearly, we could only embed relevance propagation into the second or third component [19]. Since the search engine indexes the Web offline, and implement the search operation online, we will discuss the efficiency of relevance propagation for the online case and offline case respectively.

5.1 Online Complexity

Due to the algorithm descriptions, all the relevance propagation models have two kinds of computations. The first one is to retrieve the relevant pages and rank them by relevance weighting functions. Actually this is also needed by existing search engines. The second is the additionally-introduced complexity, including working set construction, relevance propagation and so on. This will be the major concern when integrating these models into the search engines. In this regard, we will focus on the analysis of these additional computations in this section. According to the model formulation and the implementation issues, we can get the following estimations on the online complexity of the relevance propagation models. Note that the time complexity we estimate here is for one query.

1. For each step of iteration in the score-level models (HS, PSH, and Slash-based models), we need to propagate the relevance score of a page along its in-link or out-link in the sub graph of the working set. Note that the source and destination pages of the hyperlink should be both in the working set, and so the average numbers of in-links and out-links per page are equal to each other. We denote this number by l. If we further use c_h to indicate the time complexity of propagating an entity from a page to another page along hyperlinks, we can get that the complexity of each step of iteration in the score-level models is wlc_h. Where w is the size of the working set. If it takes t iterations for the propagation to converge, the overall complexity will be $twlc_h$.
2. Similar to the analysis of the score-level models, we can get the complexity of the term-level models (HT, PTH, and Slash-based models) is all $twlc_h$.

5.2 Offline Complexity

Since a real search engines should handle hundreds of queries per second [2, 27], it will be very difficult to implement these propagation techniques online. So offline implementation is much more preferred if we want to apply them in real-world applications. Search engines usually build offline invert and forward indices to store the information of each term (including frequency, position and so on) in web pages [2, 27]. Then it is easily understood that term-level propagation models can well match this mechanism and we only need to refine the offline index files. To illustrate it, let us take the ST method for example. Suppose the parent pages of page p contain a particular word, and we need to propagate the occurrence frequency of this word to page p. If p already contains this particular word, we only need to modify its frequency; while if p does not contain the word, we need to add its ID to the forward index [27] of page p, and then update its term frequency. Comparatively, the score-level propagation models could hardly be integrated into search engines, because scores do not exist in the offline indices but are dependent on the online relevance ranking algorithm used in the search engine.

6 Conclusions

In this paper, we conducted a comprehensive study on relevance propagation models in web information retrieval. A new idea for using number of slashes in the URL in the relevance propagation process was proposed (more number of slashes in the URL, the less valuable web page is). It is consistent with the findings by Najork and Wiener [16], and Ricardo Baeza-Yates and Carlos Castillo [2]. To evaluate the proposed model, the Letor 3.0 web test collection was used in the experiments. The following conclusions are drawn from our study:

1. Generally speaking, relevance propagation can boost the performance of web information retrieval.
2. Using number of slashes in the URL in the propagation process can boost the accuracy of the relevance propagation.
3. The Slash-based propagation model outperforms the hyperlink and popularity-based propagation models (PSH, PTH, HS, and HT models).
4. Our model has two methods, ST and SS, but the former is more effective and efficient than the latter.
5. Unlike the score-level models, the offline implementation of term-level models is possible. In other words, term propagation is more feasible for real-world implementation than relevance score propagation.

There are two interesting directions for further research:

1. Other than the neighbour sets derived from the explicit link structure of the Web, we can also define other types of neighbours. In general, propagation models allow us to define any set of documents with a specific characteristic as a neighbour set. As an example, we can define the set of pages with similar content as a neighbour set [17]. It is interesting to see if exploiting these types of neighbours can further improve the retrieval accuracy.
2. We want to explore effects of splitting Web page to different streams with possibly different degrees of importance on relevance propagation accuracy. That is, there is a global set of labelled streams, and the text of each Web page is split between these streams.

References

1. Alam, M.H., J. Ha, and S. Lee, Novel approaches to crawling important pages early. Knowledge and Information Systems, December 2012. 33(3): 707-734.
2. Baeza-Yates, R., Castillo, C., Crawling the Infinite Web, Journal of Web Engineering, 6(1) , 2007, 49-72.
3. Baeza-Yates, R. & Ribeiro-Neto, B. Modern Information Retrieval. ACM Press/Addison Wesley, 1999.
4. Brin, S., Page, L., The Anatomy of a Large Scale Hypertextual Web Search Engine, Proc. 7th WWW, 1998.
5. Chen, Y.-L. and X.-H. Chen, An evolutionary PageRank approach for journal ranking with expert judgements Journal of Information Science, June 2011;. 37(3), 254-272.

6. Golshani, M.A, ZarehBidoki, A.M, IECA: Intelligent Effective Crawling Algorithm for Web pages, International Journal of Information & Communication Technology Research (IJICTR).
7. Gong, Z., L.H. U, and C.W. Cheang, Web image indexing by using associated texts. Knowledge and Information Systems, August 2006. 10(2), 243-264.
8. Haveliwala, T., Topic-Sensitive Pagerank, Proc. of the 11th WWW, 2002.
9. Huberman, B.A., et al., Strong Regularities in World Wide Web Surfing. Science, April 1998. 280(5360), 95-97.
10. Jarvelin, K. & Kekalainen, J. Comulated Gainbased Evaluation of IR Techniques. ACM Transactions on Information Systems, 2002, 20(04), 422–446.
11. Jiang, L., C. Li, and Z. Cai, Learning decision tree for ranking. Knowledge and Information Systems, July 2009. 20(1), 123-135.
12. Kwon, S., Y.-G. Kim, and S. Cha, Web robot detection based on pattern-matching technique. Journal of Information Science, February 27, 2012. 38(2), 118-126.
13. Lewandowski, D., A three-year study on the freshness of web search engine databases Journal of Information Science, December 2008. 34(6), 817-831
14. Mousakazemi, E., Saram, M.A., ZarehBidoki, A.M, Popularity-based relevance propagation, Journal of Web Engineering, 2012, 1(4), 350-364.
15. Mukherjea, S., Discovering and analyzing World Wide Web collections. Knowledge and Information Systems, March 2004. 6(2), 230-241.
16. Najork,. M., Wiener, J., Breadth-First Search Crawling Yields High-Quality Pages, in 10th International conference World Wide Web, 2001.
17. O. Kurland and L. Lee. Pagerank without hyperlinks: structural re-ranking using links induced by language models. In Proceedings of ACM SIGIR, 2005, 306–313.
18. Page, L., Brin, L., Motwani, R., Winograd, T., The PageRank Citation Ranking: Bringing Order to the Web, 1998, Technical report, Stanford University, Stanford, CA.
19. Qin, T., Liu, T. Y., Zhang, X. D., Chen, Z., & Ma, W. Y. A study of relevance propagation for web search. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, 408–415.
20. Qin, T., Liu, T., Xu, J. & Li, H. Letor: A benchmark collection for research on learning to rank for information retrieval. Information Retrieval Journal, 2010, 346-374.
21. Rosset, S., C. Perlich, and B. Zadrozny, Ranking-based evaluation of regression models. Knowledge and Information Systems, August 2007. 12(3), 331-353.
22. Rosa, K.D., V. Metsis, and V. Athitsos, Boosted ranking models: a unifying framework for ranking predictions. Knowledge and Information Systems, March 2012. 30(3), 543-568.
23. Robertson, S., Jones, K., Relevance Weighting of Search Terms, Journal of the American Society of Information Science, 129-146.
24. Robertson, S., Overview of the Okapi Projects, Journal of Documentation, Vol. 53, No. 1, 1997, 3-7.
25. Salton, G., Buckley, C., Term weighting approaches in automatic text retrieval, Information Processing and Management, 1988, 24(5), 513-523.
26. Shakery, A. & Zhai, C. X. Relevance Propagation for Topic Distillation UIUC TREC 2003 Web Track Experiments. In Proceedings of the TREC Conference, 2003.
27. Shakery, A. & Zhai, C. X. A probabilistic relevance propagation model for hypertext retrieval. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM), 2006, 550-558.
28. Shchekotykhin, K., D. Jannach, and G. Friedrich, xCrawl: a high-recall crawling method for Web mining. Knowledge and Information Systems, November 2010. 25(2), 303-326.

29. Song, R., Wen, J., Shi, S., Xin, G., Liu, T., Qin, T., Zheng, X., Zhang, J., Xue, G., Ma, W., Microsoft Research Asia at Web Track and Terabyte Track of TREC 2004, Proc. the 13th TREC, 2004.
30. S. Pandey and C. Olston, “User-centric Web crawling,” in 14th international conference on World Wide Web, 2005.
31. Wang, B., et al., Query-dependent cross-domain ranking in heterogeneous network. Knowledge and Information Systems, January 2012.
32. Xia, F., et al., Ranking with decision tree. Knowledge and Information Systems, December 2008. 17(3), 381-395.
33. ZarehBidoki, A., Yazdani, N., DistanceRank: An intelligent ranking algorithm for web pages, Information Processing and Management, 2008, 44(2).