A Thesis for the degree of Master

A Rapid Egocentric Search Scheme Using Authority Estimation in Blog Space

Yoonjae Jeong

School of Engineering
Information and Communications University
2007
A Rapid Egocentric Search Scheme Using Authority Estimation in Blog Space
A Rapid Egocentric Search Scheme Using Authority Estimation in Blog Space

Advisor: Professor Dongman Lee

by

Yoonjae Jeong
School of Engineering
Information and Communications University

A thesis submitted to the faculty of Information and Communications University in partial fulfillment of the requirements for the degree of Master of Science in the School of Engineering

Daejeon, Korea
Approved by

______________________________ (signed)
Professor Dongman Lee
Major Advisor
A Rapid Egocentric Search Scheme Using Authority Estimation in Blog Space

Yoonjae Jeong

We certify that this work has passed the scholastic standards requested by the Information and Communications University as a thesis for the degree of Master

June 11, 2006

Approved:

Chairman of the Committee
Dongman Lee, Professor
School of Engineering

Committee Member
Minsoo Hahn, Professor
School of Engineering

Committee Member
Sung-Hyon Myaeng, Professor
School of Engineering
Abstract

A blog is a new media that is receiving a lot of attention. Its links enable us to get a hold of social relations between bloggers in a blog space, and the relations reflect bloggers’ interests. Therefore, the ability to search documents in linked blogs is significant for bloggers. An egocentric search method was proposed to search for documents in such neighboring blogs. However, it takes quite considerable time to find the most valuable documents in a user’s neighboring blogs when many blogs are linked to that user’s blog. Therefore, the purpose of our study is to improve the egocentric search speed for important documents in the neighboring blogs. To achieve this goal, we are proposing a rapid egocentric search scheme that reduces the search space to more important blogs. Our study shows that the number of neighboring blogs, which are linked to a blog with trackbacks and comments, is important for estimating the authority of blog. In the experimental results, our method was four times as fast as the egocentric search using a breadth-first search strategy in searching for the
top 5% of the most important documents in the neighboring blogs.
Contents

Abstract ........................................................................................................................................ i

Contents ....................................................................................................................................... iii

List of Tables ................................................................................................................................ v

List of Figures .......................................................................................................................... vi

List of Abbreviations ................................................................................................................ vii

I  Introduction .......................................................................................................................... 1
   1.1 Motivation ..................................................................................................................... 1
   1.2 Contribution ................................................................................................................ 3
   1.3 Thesis Organization ................................................................................................. 4

II  Related Work ......................................................................................................................... 6
   2.1 Ranking Algorithms .................................................................................................. 7
   2.2 The Egocentric Search ............................................................................................ 9
   2.3 Page Ordering Strategies in the Focused Crawling Area ....................................... 11

III The Proposed Scheme .......................................................................................................... 12
   3.1 The Concept of the Proposed Rapid Egocentric Search ...................................... 12
   3.2 Design Considerations ............................................................................................. 14
      3.2.1 Estimation of a Blog’s Authority ................................................................. 14
      3.2.2 Minimum Information Requirement and Estimation Process ............... 14
      3.2.3 Robustness against Spam ....................................................................... 14
   3.3 A Heuristic Function for Authority Score Estimation ........................................ 15
   3.3 Heuristic Egocentric Search .................................................................................. 22

IV Evaluation ............................................................................................................................ 25
   4.1 Experimental Environment and Tools ................................................................. 25
4.1.1 Overview of Experiment Tools ...............................................................26
4.1.2 Blog Crawler ...........................................................................................28
4.1.3 Link Extractor ..........................................................................................30
4.1.3 Indexer .....................................................................................................31
4.1.4 Ranker .....................................................................................................32
4.1.5 Visualizer ................................................................................................34
4.2 Performance Metric ......................................................................................35
4.3 Comparison with Blind Search and Various Page-Ordering Strategies ...............................................................36
4.4 Effects of the Number of Target Documents .................................................38
4.5 Effects of Search Ranges .............................................................................39

V Conclusion ......................................................................................................41
5.1 Summary ......................................................................................................41
5.2 Future Work ................................................................................................41

References .........................................................................................................45
Acknowledgements ............................................................................................49
Curriculum Vitae ...............................................................................................50
List of Tables

Table 1. The egocentric search and centralized web search................................. 2
Table 2. Comparison with PageRank, HIT, and EigenRumor algorithm [18] .... 8
Table 3. $R^2$ of test set....................................................................................... 21
Table 4. Extraction links from gathered data ..................................................... 30
Table 5. Type of link and edge in the visualizer.................................................. 34
List of Figures

Figure 1. Combination of the egocentric search and the centralized web search . 4
Figure 2. The three egocentric search methods [31] .............................................. 10
Figure 3. The concept of the proposed egocentric search .................................. 13
Figure 4. Relations between authority, reputation, and hub scores .................. 15
Figure 5. Distribution of blogs’ a (authority score) ............................................. 16
Figure 6. Distribution of ln (a) ........................................................................ 17
Figure 7. Distribution of -1 / ln (a) ............................................................. 18
Figure 8. The number of communicating blogs and the authority score ............ 19
Figure 9. The log-scaled number of communicating blogs and the authority score ................................................................. 19
Figure 10. The flow of the proposed egocentric search scheme ..................... 23
Figure 11. Brief architecture illustration of experiment tools ......................... 27
Figure 12. Diagram of the blog crawling process .......................................... 29
Figure 13. The basic EigenRumor algorithm [18] ............................................ 33
Figure 14. Visualizer graphic representation of an ego blog and its neighbors... 35
Figure 15. Recall rates for the search space size ............................................. 37
Figure 16. Recall rates for the number of target documents ............................ 39
Figure 17. Recall rates for maximum hop distance ......................................... 40
List of Abbreviations

ICU       Information and Communications University
I Introduction

This thesis proposes and evaluates a rapid egocentric search scheme for networked blog spaces, where bloggers in a blog network can interact with each other’s blogs by making trackbacks and comments. The main goal of egocentric search in blog spaces is that users can search for appropriate documents regarding their query in neighboring blogs through a search medium. Neighboring blogs are blogs linked with the user’s blog within a limited hop distance. However, the more the neighboring blogs there are, the more time the egocentric search will spend; therefore, a major issue is to reduce egocentric search time.

1.1 Motivation

Blogs allow users to record and publish their thoughts, personalities and life stories in cyberspace [22]. In addition, blog contents are easily shared with other people because they are interconnected with each other. Some recent studies show that individual blogs are transforming into communities of bloggers that share a common set of interests [26], [33]. For example, people use blogs for exchanging interesting information with acquaintances [3]. Therefore, it is important to allow a user to search through blogs closely related to the user’s interest.

There are two types of approaches for searching documents in neighboring blogs which share common interests with a user: egocentric search [31] and centralized web search [19], [20]. The egocentric search aims to find documents in blogs closely related to the user’s blog. However, it has two problems. First, it
is very time consuming to search for related documents when a large number of blogs are involved in a user’s blog network. It is a waste of time to rummage for valuable documents in all neighboring blogs. Second, it does not provide a list of the retrieved documents sorted according to their importance. Therefore, the user needs to navigate through all of the retrieved documents one-by-one to obtain the important documents. On the other hand, centralized web search saves all of the web pages, and determines the relevance of the pages. This makes it possible for a user to obtain an ordered list of documents based on their search criteria quickly. However, the retrieved results are usually only a few popular pages from the entire Web [15], and the crawling cycle of the search server for a blog becomes longer if the blog is not popular [19]. Due to these reasons, search results are often undesirable and not directly related to the user’s interests, or the user is unable to search for interesting documents in his or her neighboring blogs when the blogs do not receive a lot of attention relative to the rest of the Web.

Table 1. The egocentric search and centralized web search

<table>
<thead>
<tr>
<th></th>
<th><em>Egocentric Search</em></th>
<th><em>Centralized Web Search</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Result</td>
<td>Personally Interested Blog Document</td>
<td>Generally Interested Blog Document</td>
</tr>
<tr>
<td>Search Speed</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>
1.2 Contribution

In this paper, we propose a new scheme called heuristic egocentric search that combines the benefits of both the egocentric search and the centralized web search. This is realized by limiting the egocentric search space to only valuable and relevant blogs. The proposed approach improves the speed of the egocentric search process and the quality of retrieved documents. For this, we propose a heuristic function that estimates the authority value of a blog. The term “authority” refers to the importance of a blog. We use the estimated authorities instead of those established by other ranking algorithms. If we use the latter, we cannot trace the dynamic changes of blog space due to complexities of the algorithms. The proposed heuristic egocentric search method arranges the order of neighboring blogs using an estimation function, and limits the search space to blogs having high authority values. This process is similar to the page-ordering approach in the focused crawling area in the aspect of reducing target space [7]. However, because the page-ordering approach focuses only on gathering general web pages, a more appropriate blog ordering strategy is needed when we consider the egocentric search ranges, blog link types, and the ability of blogs to provide information.

The contributions of this paper are two-fold. First, we find that the number of neighboring blogs, which are linked to a blog with trackbacks and comments, is important for estimating the value of the blog. A trackback is a mechanism for communication between blogs [2]. It enables bloggers to refer their own documents to other bloggers’ documents. Comments are used for bloggers to add their opinions on other bloggers’ documents. Second, we confirm that the
The proposed scheme using a best-first search strategy achieves high performance in terms of accuracy in retrieving interesting documents while decreasing search time. Our experimental results show that the proposed method reduces the number of visited blogs by 38% compared to the egocentric search using a breadth-first search strategy when the top 5% most interesting documents are retrieved from neighboring blogs. In addition, the proposed approach is around four times faster than the original egocentric search.

Figure 1. Combination of the egocentric search and the centralized web search

1.3 Thesis Organization

The rest of the thesis is organized as follows. The related work is presented in Chapter 2. Chapter 3 presents the proposed heuristic function that improves the performance of the egocentric search method. In Chapter 4, we discuss the environmental settings, and then analyze the results of our experiment. Finally,
Chapter 5 concludes the thesis.
II Related Work

Blogs have recently begun receiving a lot of attention. Therefore, research on blog space has only recently been initiated. The early blog studies focused on analyzing the characteristics of blog space and classifying them. Blood classified blogs into three basic types: filters, personal journals, and notebooks [10]. Krishmanurthy broke down blogs into four basic types based on two dimensions: personal vs. topical and individual vs. community [25]. This trend of blog classification is still continuing. Lindahl and Blount divided blog styles into two types; filters and journals [27], and Herring et al. distinguished three types of blogs; personal journals, filters, and knowledge logs [21].

However, later studies have also included the contents and social aspects of blog space. Bar-Ilan divided blogging activities into three types: community blogs, group blogs, and a mixture of group and community blogs [9]. In addition, Nardi et al. presented preliminary results from an ethnographic exploration of blogs conducted initially [28]. The focus of Nardi et al.’s research is on understanding blogging as a new form of personal expression and communication. They also presented findings on blogging motivations such as personal journaling. Kumar et al. identified bloggers’ demographics, friendships, and activity patterns by analyzing the structure and content of more than one million blogs worldwide [26].

In addition to the blog content, some studies show that blog spaces are taking on the features of community space. According to Kumar et al.’s study [26], bloggers have been moving steadily towards more community-oriented activity, and they also suggested that there are three layers in blog space: the
individual blogger, a web of friendship, and blog communities. Rosenbloom said that blog is a powerful tool to establish an online community [33], and Tepper made an assertion that blogs might soon become our primary way of interacting with one another online [35].

In this thesis, we focus on a blog search method that utilizes the social aspect of the blog space. This chapter first gives ranking methods for individual blog spaces, and then we review a new methodology for searching within a blog network.

2.1 Ranking Algorithms

The PageRank [12] and HIT [24] algorithms are currently the two most popular algorithms used to evaluate web documents. However, these approaches do not focus on ranking the owner of a document but the document itself. They are not proper for evaluating bloggers and their blogs. The EigenRumor algorithm is used for evaluating members’ contribution in a cyberspace community [18]. The EigenRumor algorithm uses authority and hub scores for appraising the members’ contribution to the community, and reputation scores for objects that the members produce. The authority score indicates how often a member creates valuable objects, and the hub score represents how often a member evaluates them. In other words, a member having a high authority score creates many interesting objects, and a member with a high hub score evaluates many valuable objects. The EigenRumor algorithm also calculates the reputation score of an object using the number of evaluations by members who have high hub scores. The reputation score becomes the value of the document. This can
also be applied to blog space. When we apply EignenRumor algorithm to blog space, the members become bloggers, and the objects are documents. The EigenRumor algorithm as applied by Fujimura et al. [18] uses only trackback links, but we use both trackbacks and comments because bloggers make comments much more frequently than trackbacks. Although this algorithm is a good mechanism to calculate the values of blogs, they need to analyze all of the data in the blog space. It should have a long iteration process to all of the blogs, as well. These algorithms require a lot of computing resources because blog documents are frequently updated.

Table 2. Comparison with PageRank, HIT, and EigenRumor algorithm [18]

<table>
<thead>
<tr>
<th>Entities</th>
<th>PageRank</th>
<th>HITS</th>
<th>EigenRumor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link types</td>
<td>Evaluation (E)</td>
<td>Evaluation (E)</td>
<td>Evaluation (E) Provisioning (P)</td>
</tr>
<tr>
<td>Scores</td>
<td>Authority (a)</td>
<td>Hub (h)</td>
<td>Reputation (r)</td>
</tr>
<tr>
<td>Algorithm</td>
<td>[ a = \left( \frac{d}{N} \right)^{\frac{1}{r}} \left( 1 - d \right) + \alpha ]</td>
<td>[ h = E \alpha ]</td>
<td>[ r = \alpha (P r) + (1 - \alpha) E h ]</td>
</tr>
<tr>
<td></td>
<td>[ a = E h ]</td>
<td>[ a = E h ]</td>
<td>[ a = P r ]</td>
</tr>
<tr>
<td></td>
<td>[ r = r ]</td>
<td>[ r = r ]</td>
<td>[ h = E r ]</td>
</tr>
</tbody>
</table>
2.2 The Egocentric Search

The egocentric search method [31] provides mechanisms to search for documents using the concept of “Information and Communicate Activities Navigation (ICAN).” Ohmukai et al. [31] say that the process of creating knowledge is when “People establish relationships with other people, then collaborate with each other to create new information, and finally present themselves as donors of new information.” In this process, it is important to search the information created by people who communicate with a user. They applied their concept to a blog space, and proposed three search methods: Relative Chain Search (Figure 2a), Co-citation Search (Figure 2b), and Keyword Search (Figure 2c). Relative Chain Search discovers documents linked with trackbacks from the documents related to a user’s blog. Co-citation Search finds documents that link to the same documents as the user’s blog links. Finally, Keyword Search finds documents by matching keywords in the user’s neighboring blogs. The results of the egocentric search shows that closely connected blogs to a blog have similar inclinations to the original blog [30].
In the egocentric search, the search range is limited to neighboring blogs within a 1-hop distance. A user is just able to find only a few documents in the blogs within a 2-hop distance in the case of Relative Chain Search. However, the user may need to search the neighboring blogs within 2- or 3-hop distances from his or her blog. This is because the neighbors of the user’s neighboring blogs may belong to a blog community sharing interests with the user, and receive the confidence of the members of the community. When we search for documents in neighboring blogs more than two hops away, the number of target blogs increases exponentially. This causes a huge increase of egocentric search
time. Due to the fact that the original egocentric search does not evaluate the value of retrieved documents, its results are not sorted in order of the importance of retrieved documents, either.

2.3 Page Ordering Strategies in the Focused Crawling Area

Ordering pages is important in the focused crawling area because indexing the entire Web is quite expensive work. Therefore, it is relevant to index at least the most important pages. Baeza-Yates et al. [7] performed some experiments on and compared various page-ordering strategies. The main strength of these strategies falls on crawling general web pages in the entire Web. Therefore, the strategies cannot be applied to blog ordering in our egocentric search due to the following differences between the web crawling and the egocentric search. First, the search scope of the egocentric search is the only neighboring blogs around a blog. Second, there are two important types of hyperlinks in blog spaces, which are trackbacks and comments, but the existing page-ordering strategies treat them the same. Finally, current blogs can provide rich information using an RSS (Really Simple Syndication) [5] feed. Therefore, a more effective strategy can be obtained when the egocentric search are provided with appropriate information from blogs.
III  The Proposed Scheme

The main goal of the proposed scheme is to find important documents in a user’s neighboring blogs rapidly. It seems reasonable to suppose that one would look for documents in blogs having high authorities to retrieve highly reputed documents. By applying this benefit of centralized web search engines, which already knows the authority values of blogs, the egocentric search process can reduce the search space from all of neighboring blogs to only valuable blogs. This saves a significant amount of search time for important documents in neighboring blogs.

In this chapter, we first illustrate the concept of the proposed rapid egocentric search scheme, and then discuss what should be considered in estimating the value of neighboring blogs. We finally suggest estimation functions for blogs, and then describe the proposed scheme in detail.

3.1 The Concept of the Proposed Rapid Egocentric Search

Figure 3 illustrates the concept of the proposed egocentric search. The ego blog site is a user’s blog, and neighboring blogs are the blog sites with which a user communicates directly or indirectly. In the case of the egocentric search using a breadth-first search strategy, blogs are visited in order of ①, ②, ③, ④, ⑤, ⑥, and ⑦, but the proposed strategy just visits ②, ⑤, and ⑥.
Figure 3. The concept of the proposed egocentric search

However, the authority value of a blog is not always known. If a blog is not popular relative to the whole blog space, centralized web search engines do not gather and evaluate it. Some commercial blog services even prevent crawlers of web search engines from visiting their blog sites. For these reasons, we need to estimate the values of neighboring blogs on the basis of the limited information provided by an individual blog. It enables us to know the authority value of a blog, and to perform the enhanced egocentric search retrieving high quality documents in less search space.
3.2 Design Considerations

3.2.1 Estimation of a Blog’s Authority

It is necessary to estimate the value of a blog during the proposed egocentric search process. Although the central web search can evaluate the authorities of blogs, it is not suitable for calculating and updating a blog’s authority frequently. This is due to an infinite amount of blog documents in the whole Web and the limitations of computing resources like network bandwidth and disk space.

3.2.2 Minimum Information Requirement and Estimation Process

We need to identify the minimum set of data for estimating a blog’s authority, and should simplify the evaluation process to estimate a blog’s authority quickly. The modern web ranking algorithms like the PageRank, HIT and the EigenRumor algorithm require all of the blog documents and their hyperlinks. They also use much iteration to evaluate a blog. Such heavy data requirements and evaluation processes cause the long evaluation time. These algorithms cannot cover the dynamic changes of blog space; blogs are very frequently updated, and interconnected with trackbacks and comments.

3.2.3 Robustness against Spam

Our proposed egocentric search should be robust against spam links. Spam is one of the top challenges for web search engines. It is because the modern web ranking algorithms are based on the analysis of hyperlinks. Since spam can make us misjudge the value of a blog, it is important to minimize the effect of
3.3 A Heuristic Function for Authority Score Estimation

When we use the EigenRumor algorithm, the authority score of a blog depends on the number of documents with high reputation scores. The number of high hub score blogs linked to a document also determines a reputation score. From these factors, we can derive that there is a relationship between the authority score of a blog and the number of its neighboring blogs.

Figure 4. Relations between authority, reputation, and hub scores

We conjecture that the authority score of a blog is high when it has many inbound links, but we use the number of the neighboring blogs linked to a blog instead. This is because the number of inbound links is very vulnerable to spam links. Spam bloggers are a few portions of neighboring blogs around a blog, so the number of neighbor blogs is less affected by spamming. The data size of the
number of neighboring blogs is also very small if blogs yield the number with various methods such as XML-RPC [36].

Before turning to observe the relation between a blog’s authority score and the number of communicating blog, we must draw attention to the distribution of authorities of blogs. If the authority scores are not normally distributed, a statistical analysis such as the linear regression cannot be applied [32]. From previously gathered data from a blog service, Egloos [1], we found that the blogs’ authority scores, which are calculated by the EigenRumor algorithm, are not normally distributed. Figure 5 shows the distribution. In this figure, $x$-axis indicates a blog’s authority score, and $y$-axis is the number of blogs which have the score.

![Figure 5. Distribution of blogs’ $a$ (authority score)](image)

16
When we look at the distribution of blogs’ authority scores, most of the blogs have very small scores. To transform this to the normal distribution, we used log-scaled authority scores instead of pure authority values. Figure 6 represents the transformed distribution.

The distribution of $\ln(a)$ in Figure 6 is closer to the normal distribution than that of pure authority scores, but the center of the distribution is in the right side of the graph. Therefore, by using $-1/\ln(a)$, we got the distribution which are closer the normal distribution than previous distributions. The graph is presented in Figure 7.
Figure 7. Distribution Distribution of $-1 / \ln (a)$

From gathered data, we find the tendency that the negative inverse of log-scaled authority scores has a linear relationship with the logarithmically scaled number of neighboring blogs, which are directly linked to the blog. In Figure 9, the $x$-axis shows the log-scaled number of the neighboring blogs that are linked to a blog with comments, and the $y$-axis indicates that of trackbacks. The $z$-axis is the negative inverse of the logarithmically scaled authority score of the blog.
Figure 8. The number of communicating blogs and the authority score

Figure 9. The log-scaled number of communicating blogs and the authority score
From this observation, we define the relationship between the number of neighboring blogs and its authority score in (1). In this equation, \( n_c \) and \( n_t \) are the number of neighboring blogs linked to a blog using comments and trackbacks, respectively, and \( a \) is the authority score of the blog.

\[
-\frac{1}{\ln(a)} = \beta_0 + \beta_1 \times \ln(n_c) + \beta_2 \times \ln(n_t)
\]  

(1)

We obtain the constants \( \beta_0 \), \( \beta_1 \) and \( \beta_2 \) using the multiple linear regression method [22] on the training data. The graphed surface in Figure 9 is the result of regression. From (1), we derive the equation below that represents an estimation of a blog’s authority score. \( h(n_c, n_t) \) is a heuristic function to estimate the authority score of a blog.

\[
h(n_c, n_t) = \begin{cases} 
1 & \text{if } n_c > 0 \text{ and } n_t > 0 \\
\frac{1}{e^{-\beta_0 + \beta_1 \ln(n_c) + \beta_2 \ln(n_t)}} & \text{if } n_c > 0 \text{ and } n_t = 0 \\
\frac{1}{e^{-\beta_0 + \beta_2 \ln(n_t)}} & \text{if } n_c = 0 \text{ and } n_t > 0 \\
0 & \text{otherwise}
\end{cases}
\]  

(2)

\[
\beta_0 = 0.050927545695806570 \\
\beta_1 = 0.015686308847722800 \\
\beta_2 = 0.004685314635476029
\]

To verify our heuristic function, we examined it for nine test sets that were
randomly selected from gathered blog data. $R^2$ (Coefficient of Determination) is the proportion of variability in a data set [32]. In (3), $SS_T$ is the total sum of squares, $SS_R$ is the regression sum of squares, and $SS_E$ is the sum of squared errors.

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T}$$

$$SS_T = \sum (y_i - \bar{y})^2, \quad SS_R = \sum (\hat{y}_i - \bar{y})^2, \quad SS_E = \sum (y_i - \hat{y}_i)^2$$

If $R^2$ is close to one, it means that the heuristic function fits the data set well. Table 3 shows test sets and their $R^2$ values. The $R^2$ values are all very close to one, and the heuristic function is acceptable for estimating authority scores for blogs.

<table>
<thead>
<tr>
<th>Set</th>
<th># of Sample Blogs</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set</td>
<td>1171</td>
<td>0.9539</td>
</tr>
<tr>
<td>Test Set 1</td>
<td>1153</td>
<td>0.9523</td>
</tr>
<tr>
<td>Test Set 2</td>
<td>1160</td>
<td>0.9502</td>
</tr>
<tr>
<td>Test Set 3</td>
<td>1139</td>
<td>0.9504</td>
</tr>
<tr>
<td>Test Set 4</td>
<td>1162</td>
<td>0.9490</td>
</tr>
<tr>
<td>Test Set 5</td>
<td>1171</td>
<td>0.9453</td>
</tr>
<tr>
<td>Set</td>
<td># of Sample Blogs</td>
<td>$R^2$</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Test Set 6</td>
<td>1155</td>
<td>0.9522</td>
</tr>
<tr>
<td>Test Set 7</td>
<td>1205</td>
<td>0.9504</td>
</tr>
<tr>
<td>Test Set 8</td>
<td>1163</td>
<td>0.9528</td>
</tr>
<tr>
<td>Test Set 9</td>
<td>1193</td>
<td>0.9073</td>
</tr>
</tbody>
</table>

### 3.3 Heuristic Egocentric Search

To improve the egocentric search time, it is necessary to reduce the search space with an algorithm such as greedy best-first search [34]. It expands its state to the node closest to a goal based on a heuristic function. The proposed egocentric search process gathers information about neighboring blogs in query time, and estimates their authority score. The neighboring blogs are visited in turns, and we look for documents that are suited to a user’s query. Figure 10 illustrates an execution flow of the proposed scheme.

As Figure 10 illustrates, we first list the user’s neighboring blogs, and get the number of blogs linked using a trackback or a comment. Then we get the estimated authority scores of the listed blogs, and draw them in order by their score. The estimated score is calculated using the heuristic function in (2). The detailed procedure is below.
Step 1. Input a user’s ego blog.

Step 2. Get the user’s neighboring blogs

Step 3. Enqueue the user’s neighboring blogs into the queue.

Step 4. Is the queue empty?

Step 5. Dequeue a neighboring blog from the queue.

Step 6. Calculate the estimated authority score of the blog.

Step 7. Enqueue the blog into the priority queue with the estimated score.

Step 8. Is the queue empty?

Step 9. Is the cumulative number of neighboring blogs got from the priority queue bigger than the size of search space?

Step 10. Dequeue a blog from the priority queue.

Step 11. Search documents in the blog.

Step 12. Is the distance between the user’s ego blog and this blog longer than the max hop-distance?

Step 13. Get the neighboring blogs of the blog.

Step 14. Enqueue the neighboring blogs into the queue.

Stop

Figure 10. The flow of the proposed egocentric search scheme
We first get a user’s blog, and put its neighboring blogs into a queue (steps 1~3). Next, we ensure that we have blogs in the queue (step 4), then obtain a blog from the queue (step 5), estimate its authority score using the heuristic function (step 6), and put it into a priority queue in which the estimated authority scores determine the order of the blogs (step 7). The blogs within the priority queue determine the search space. To find documents, we get a blog from the priority queue (step 10) and search documents in the blog (step 11). We check whether the distance between a user’s blog and its neighboring blog has reached the maximum search distance or not (step 12). If the distance is less than the maximum distance, we get the neighboring blogs of a blog and put them into the queue (steps 13~14). We repeat this process until the queue is empty (starting over at step 4) or we have searched a fixed number of blogs (step 9).
IV Evaluation

In this chapter, we evaluate the performance of the proposed egocentric search scheme through experiments. To conduct the experiments, a experimental environment was created to perform like a real blog space, and the egocentric search applications were then written on that environment. Then, we compared the original egocentric search method and the proposed heuristic egocentric search method.

In the following sections, first experimental environment is described, and then the performance metrics are stated. The last section compares the performance of the two search methods.

4.1 Experimental Environment and Tools

For experimentation of the proposed egocentric search scheme, we create an experimental environment to emulate a real blog space and egocentric search applications. Then, we compared the egocentric search using the existing page-ordering methods with the proposed rapid egocentric search. The experimental data consisted of 389,695 documents from 28,737 blogs gathered from Egloos [1].

To measure the performance of our heuristic function and search method, we set a search range to the neighboring blogs that are at most three hops away from a user's blog. We then search the documents with the top 1~10% reputation score calculated by the EigenRumor algorithm for each keyword. These documents are more valuable in the search space. The keywords used were “animation,” “comics,” “literature,” and “music.” Next, we compared the results with those of
the Keyword Search method in the egocentric search. We chose 30 blogs per keyword, and made them the users’ blogs for the egocentric search.

4.1.1 Overview of Experiment Tools

Our team developed the tools for use in the experiments. The tools were constructed on the Linux System using Java programming Language. The gathered blog data was stored in MySQL DBMS [3]. We also developed egocentric search programs to use in this experimental environment. One program emulated the proposed heuristic egocentric search method, and the other programs performed the egocentric search method using basic search strategies and various focused crawling strategies. The egocentric search program visited the neighboring blogs in turn, and found suitable documents for the given keyword.

The experiment tools provided several modules such as a blog crawler, link extractor, indexer, ranker, and visualizer which are the basic building blocks for conducting an experiment in blog space. The architecture is illustrated in Figure 11.
The role of each module is as follows:

- The blog crawler is used for gathering real blog data from the Internet.
- The database and file system are used for storing the gathered blog data and the results from other modules.
- The link extractor parses and extracts links from gathered blog documents.
- The indexer performs indexing of gathered documents according to keyword.
- The ranker evaluates the value of a blog and documents within the blogs.
- The visualizer produces a visual representation of the networks between
blogs in graphic format.

4.1.2 Blog Crawler

The blog crawler is used for gathering real blog data from the Internet. The blog crawler sets a blog as the seed, gathers the documents within that blog, and then visits the blogs that make trackbacks and comments to the documents of the blog in turn. In addition, the crawler visits blogs that the user bookmarks. The collected documents are stored in the file system and database. Figure 12 below shows the process of gathering blog documents from the Internet.
Figure 12. Diagram of the blog crawling process

In this process, the user inputs the crawling period and the root blog site, or seed, and crawling depth (step 1~2). The crawling depth is the maximum hop distance between the seed blog and a target blog. The blog crawler extracts the URLs of blogs linked from the blog with bookmarks, and put them into the crawling navigation queue (step 6, 8). The crawling navigation queue uses a FIFO (first-in, first-out) queue for saving a blog list that the crawler should visit. Then, the blog crawler navigates the documents of the blog in the given period,
and puts the blogs linked to the document into the crawling navigation queue (step 8). The blog crawler repeats this process until the crawling navigation queue is empty (step 3). This process is similar to a depth-first tree search.

The blog visited by the blog crawler is saved in the crawling history list (step 4). The blog crawler only puts the original blog, not including the crawling history list, into the crawling navigation queue (step 7). It prevents the crawling process from falling into infinite loops.

4.1.3 Link Extractor

The link extractor analyzes the html files of the gathered blog documents, and extracts bookmark, trackback, and comment links from them. Based on this data, the link extractor gets the connection information between documents, between documents and blogs, and between blogs. It first gets the bookmark links from the main blog page, and then extracts the trackback and comment links from each blog document.

Trackback links are the connections between a document and a document, and comment links are the connections between a document and a blogger. Therefore, we should convert those links to the connection between a blogger and a blogger to apply our proposed scheme. We replace the document as the owner of document and make the links between bloggers. Table 4 below shows the converting process of different types of links.

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Trackback</th>
<th>Comment</th>
<th>Bookmark</th>
</tr>
</thead>
</table>

Table 4. Extraction links from gathered data
### 4.1.3 Indexer

The *indexer* makes an index of gathering documents by keyword. The indexing process determines to what extent a document is relevant to the given keyword. This becomes the basis to look for the document when a user types in a query.

The *indexer* first extracts the contents from the document by parsing the html file. Then, it extracts terms using an n-gram algorithm. In n-gram, we set n from 2. We calculate the relevant rate of terms and documents with $TF \times ITF$ approach. $TF$ is the term frequency, and $ITF$ is the inverse term frequency. According to the $TF \times ITF$ approach, the relevant rate of a document for search term is proportionate to the frequency of term ($TF$) in the document, but the rate is inversely proportionate to the frequency of documents having the term ($ITF$). From this information, the importance of a term in a document is represented in Equation 4:
\[ W_{ik} = F_{ik} \times \log_2 \left( \frac{N}{\text{DocFreq}_k} + 1 \right) \]

- \( W_{ik} \): the importance of a term \( k \) in a document \( i \)
- \( F_{ik} \): the frequency of term \( k \) in a document \( i \)
- \( N \): the total number of documents in the database of simulator
- \( \text{DocFreq}_k \): the frequency of document including term \( k \)

### 4.1.4 Ranker

The ranker evaluates the value of blogs. The scores are determined with the EigenRumor algorithm, and the evaluation factors are based on the links, which are extracted by the link extractor.

In the EigenRumor algorithm, the provision matrix \( P=[p_{ij}] \) represents the provision relationships between blog \( i \) and document \( j \). If the blog \( i \) has the document \( j \), \( p_{ij} \) is 1, otherwise, \( p_{ij} \) is 0. A similar matrix also expresses the evaluation between blogs. When a blogger \( i \) makes a trackback or comment to a document \( j \), we regard it as evaluating the document. In this case, the evaluation \( e_{ij} \) is 1, otherwise it is 0. An evaluation matrix \( E=[e_{ij}] \) can be constructed with these evaluation values.

Each blog has an authority score and hub score, and each document has a reputation score. These scores are notated as \( a=[a_i], h=[h_i], \text{ and } r=[r_j] \) vectors. The \( a_i \) is blog \( i \)'s authority score, \( h_i \) is the hub score, and \( r_j \) is document \( j \)'s reputation score. The process of the basic EigenRumor algorithm can be seen in Figure 13 below. In this figure, the superscripts of authority, hub, and reputation score vectors mean the iteration count of this algorithm.
\[ r^{(0)}_a = (1, L, 1)^T \]
\[ r^{(0)}_h = (1, L, 1)^T \]

while \( r \) change significantly do

\[
\frac{r^{(k)}}{r'} = \alpha P^T r^{(k)} + (1-\alpha) E^T h^{(k)}
\]

\[
\frac{r^{(k+1)}_a}{r'} = \| r^{(k)} \|_2
\]

\[
\frac{r^{(k+1)}_h}{r'} = P \cdot r^{(k+1)}_a
\]

end while

Figure 13. The basic EigenRumor algorithm [18]

However, the basic EigenRumor algorithm has an issue in that the hub and authority scores of a blog that has many evaluations and provisioning links becomes higher because these links are not normalized. Therefore, Fujimura et al. provide provisioning and evaluation matrixes for this normalization, shown in (5) below:

\[
P' = \begin{bmatrix} p'_{ij} \end{bmatrix} \quad (i = 1 \text{K} \quad m, j = 1 \text{K} \quad n) \quad p'_{ij} = \frac{1}{\sqrt{|P|}}
\]

\[
E' = \begin{bmatrix} e'_{ij} \end{bmatrix} \quad (i = 1 \text{K} \quad m, j = 1 \text{K} \quad n) \quad e'_{ij} = \frac{e_{ij}}{\sqrt{|E|}}
\] (5)
We evaluated the value of blogs and their documents using the EigenRumor algorithm. The proposed heuristic functions are designed for searching for valuable documents whose value is evaluated by the EigenRumor algorithm.

4.1.5 Visualizer

We made a visualizer to help analyze the links between blogs and authority scores more easily. The visualizer creates a graph which shows the ego blog, and vertexes and edges represent its neighboring blogs and links. The term ‘ego blog’ means a user’s blog.

If a blog is an ego blog, its vertex color is white; otherwise the vertex colors are gray. The color-depth of the vertex is representative of the blog’s authority score created by the EigenRumor algorithm. In the edges, dotted lines indicate trackbacks, comments are dashed lines, and bookmarks are plan lines. Table 5 shows that the relationships between type of link and edge.

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Edge Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookmark</td>
<td>Plain Line</td>
</tr>
<tr>
<td>Comment</td>
<td>Dashed Line</td>
</tr>
<tr>
<td>Trackback</td>
<td>Dotted Line</td>
</tr>
</tbody>
</table>

In Figure 14 below, a graph from the visualizer shows an ego blog and its neighboring blogs in 1-hop distance.
4.2 Performance Metric

The recall rate for a given search space size is used to evaluate the performance of the egocentric search method. The recall rate is a fraction of the relevant documents retrieved from all relevant documents, and it is one of important IR (Information Retrieval) performance measures [8]. Although both recall and precision are used regularly in the IR area, our experiment uses the recall rate and the size of search space. This is because our purpose is to reduce
the search space, and at the same time, to retrieve more relevant and well-reputed documents in order to improve the egocentric search speed.

4.3 Comparison with Blind Search and Various Page-Ordering Strategies

We measured how much we could reduce the egocentric search space using the proposed scheme, and compared our method with the egocentric search using the blind search methods and the page-ordering strategies employed in the focused crawling. For comparison, we chose the breadth-first search, the depth-first search from blind search strategies, and selected Backlink-count [14] and OPIC [6] from the page-order strategies of focused crawling approaches. The breadth-first search and depth-first search are basic search strategies, and Backlink-count and OPIC showed good performance in the study of Baeza-Yates et al. [7]. In the omniscient strategy, we assumed that the actual authority score of blog was known and the proposed egocentric search scheme could query a blog’s score; if our heuristic method gets close to the performance of the omniscient strategy, it means that the proposed scheme has the best performance.

Figure 15 shows the results of this observation. The search targets are documents with the top 5% reputation scores for a given keyword in the neighboring blogs within a 3-hop distance. The $x$-axis shows the ratio of selected blogs in the whole search space, and the $y$-axis indicates the average recall rate of the samples. Figure 15 shows that the recall rate of our heuristic egocentric search increases logarithmically, and it is closer to the omniscient strategy than other strategies.
When the search space was 10%, the heuristic egocentric search showed a rate of about 73.74%, but the recall rates of breadth-first search and depth-first search were 33.06% and 25.43% respectively. This shows that the heuristic egocentric search has about two and a half times better search quality than an egocentric search using the breadth-first search strategy for the above reduced search spaces. The heuristic egocentric search yielded about an 80% recall rate in a 14% search space, while the egocentric search using breadth-first search and the depth-first search was in a 52% search space. The heuristic egocentric search found results about four times as fast as the egocentric search using the breadth-first search strategy. The heuristic egocentric search also showed similar performance to the omniscient strategy after a 40% search space size. Backlink-count showed good performance, but our heuristic method was much better.
4.4 Effects of the Number of Target Documents

We also conducted an experiment on the influence of the number of target documents. When we search very highly reputed documents, the amount of the documents is very small. However, if less highly reputed documents are included in the search target, the number of target documents becomes larger. Through an observation of the change of recall rate for the number of target documents when we fixed the size of search space, we checked the effect of the number of target documents.

Figure 16 shows recall rates when the egocentric search finds documents having the top 1%-10% reputation scores in its entire search space. To examine the effect of the amount of target documents, we set the maximum hop-distance of a search space to three, and fixed the search space size to the 20% of the whole search space. In Figure 16, the $x$-axis indicates the ratio of documents having high reputation scores that the egocentric search should find, and $y$-axis is the average recall rate of the samples. The result shows that the change of recall rates of each ordering strategy is very small regardless of the number of target documents.
We also checked the effects of the search range, which is the maximum hop distance between a user’s blog and its neighboring blogs. From the change of recall for search ranges, we can determine the search range in which our heuristic search scheme is valid. Figure 17 shows recall rates when the egocentric search finds documents of blogs in the 1- to 3-hop distance from the sample users’ blogs. We set targets to the documents having the top 5% high reputation scores in the whole search space, and the search space size to 20% of the whole search space. In Figure 17, the x-axis indicates the maximum hop distance, and y-axis is the average recall rate of the samples.

Figure 17 shows that the proposed egocentric search produces higher
performance than the other strategies, regardless of the maximum hop distance. Moreover, the recall rate of our proposed strategy became close to the omniscient results when the max hop distance was high. It was also observed that the performance of Backlink-count and OPIC increased rapidly as the maximum hop distance did. The focused crawling approaches, such as Backlink-count and OPIC, are considered efficient for gathering not neighboring blogs, but the whole blog space.

Figure 17. Recall rates for maximum hop distance
V Conclusion

5.1 Summary

In this paper, we proposed an authority estimation method and a heuristic egocentric search strategy that improve the egocentric search speed. To determine which blogs are more valuable in a user’s neighboring blogs, we developed a heuristic function that predicts the authority scores on the basis of the local information of the blog. Then, we reduced the search space to relatively valuable blogs for retrieving important documents quickly by using this function.

Our experimental results showed that the blog ordering strategy using the number of linked blogs and communication types (comments and trackbacks) produced better performance than the blind search methods and the other ordering strategies of the focused crawling approach. When our heuristic function was used, it retrieved appropriate documents at a 73.74% recall rate, even if it searched documents in only 10% of the neighboring blogs. The recall rate was about 2.23 times higher than breadth-first search. When the recall rate was fixed at 80%, the heuristic egocentric search required only 14% of the whole search space. It was also about four times faster than that of the egocentric search using a breadth-first search strategy. The proposed method also provided good performance regardless of the number of target documents and search ranges (maximum hop distance).

5.2 Future Work

Although the proposed search scheme runs faster than the original
egocentric search, there is still room to improve the egocentric search speed. First, we only considered links between blogs, without inspecting the contents of blogs. When we searched documents in a user’s neighboring blogs for daily life topics, the performance was worse than for other topics. This is because personal daily lives are not an interesting topic for most people in blog space, but familiar bloggers who know each other in private actively communicate with each other about these topics in blog space. Second, we also observed a pattern of trackback communications, which is similar to a brief survey, between bloggers: a blogger posts a short questionnaire in his or her blog, and other bloggers answer it by trackbacks. The questionnaire has many inbound trackback links, but is seldom informative. We plan to study further the forms of communications for various topics and contents of the communications between bloggers. We expect that they will make our heuristic egocentric search more elaborated and effective.

We also need to evaluate our scheme more severely beyond the reduction of search space to improve the egocentric search’s speed, which we have done in this paper. In the future, we will strengthen our evaluations in aspects of computational efficiency and user satisfaction about the search results.
BFS (Breadth-first Search) 를 사용하여 5% 이상의 개선율을 얻었습니다.

4

KAIST
References


B. W. Dearstyne, "Blogs: the new information revolution? RIM professionals have an opportunity to provide leadership and guidance in the development of policies to ensure that blogs are managed as records," *Information Management Journal*, vol. 39, no. 5, September/October 2005.


S. C. Herring, I. Kouper, J. C. Paolillo, L. A. Scheidt, M. Tyworth, P. Welsch, E. Wright, and N. Yu, "Conversations in the blogosphere: An analysis 'From the


