



Université d'Ottawa • University of Ottawa



Université d'Ottawa - University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTDOCTORALES

FACULTY OF GRADUATE AND
POSTDOCTORAL STUDIES

Yongsheng LIU

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE - DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Development of an Information Retrieval
and Distillation Agent

M. Liang

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

D. Lane

D. Neculescu

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES
SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE
AND POSTDOCTORAL STUDIES

Development of an Information Retrieval and Distillation Agent

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies in
partial fulfillment of the requirements for the degree of
Master of Science

Yongsheng Liu

Systems Science, University of Ottawa

Ottawa, Ontario, K1N 6N5, Canada

January 2003

© Yongsheng Liu, Ottawa, Canada, 2003



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 0-612-90111-4
Our file *Notre référence*
ISBN: 0-612-90111-4

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Though a large number of search engines are commercially available today, the use of most of them often involves tedious human efforts. Also, a large amount of information obtained using the existing search engines may or may not be relevant to the intended query. Furthermore, there is a lack of systematic approach to quantify the value of the information for the user's needs. In this thesis, to free the user from the drudgery of the search and to provide a basis for building personalized database for a particular topic, we develop a web search and distillation agent. To retrieve the information with higher quality, we modified the existing Term frequency vs Inverse Document Frequency (TFIDF) term weighting scheme and combined it with the Hyperlink Induced Topic Search (HITS) method to create a solution measuring both importance and relevancy of a document. To construct a dynamic graph and ensure an affordable continuous search, we propose a Sliding Window Model (SWM) which is used to control the size of the node set of a graph. To improve the intelligence of the search agent, we employ the Exponential Smoothing (ES) approach to guide the search.

Our experimental results show that the proposed web search and distillation approach with the above features is effective compared to other algorithms and models: the improved TFIDF algorithm improves the rationality of the search results; the proposed SWM can control the size of the node set as expected; the ES algorithm employed in SWM can further save computing time and help the search agent harvest the information with higher quality, and gains much more advantages compared to other methods implemented in the search agent.

Acknowledgements

I would like to thank my thesis supervisor Dr. Ming Liang for his support, patience, understanding and guidance throughout the completion of my thesis. His instrumental and insightful suggestions are the constant source of my creation.

I also would like to thank our network administrator, Mrs Bradley for her providing a private lab for me to conduct my experiments, and all my classmates who had ever helped me for their valuable discussions and suggestions.

List of Figures

Figure 2-1 Vector space model	5
Figure 2-2 Zipf's law	6
Figure 2-3 Sub-graph of a directed graph model	12
Figure 2-4 Authority	14
Figure 2-5 Hub	14
Figure 2-6 Flow chart of iterative algorithm	16
Figure 3-1 Flow chart of computing the RC value	21
Figure 3-2 Free Expansion Model	23
Figure 3-3 Sliding Window Model	25
Figure 3-4 A simple graph	27
Figure 3-5 Forecasting value of a node	34
Figure 3-6 Search path and node set controlled by the ES algorithm	36
Figure 3-7 Flow chart of the overall algorithm	37
Figure 4-1 Architecture of the search agent	42
Figure 4-2 Class diagram	44
Figure 5-1 Screen shot of the first result page from Yahoo	46
Figure 5-2 Top 10 retrieved URLs based on <i>RC</i> value	48
Figure 5-3 Top 10 retrieved URLs based on the similarity	49
Figure 5-4 Comparison of TV values (Case A)	52
Figure 5-5 Comparison of TV values (Case B)	55
Figure 5-6 Comparison of TV values (Case C)	58
Figure 5-7 Comparison of TV values (Case D)	58
Figure 5-8 Comparison of computing times (Case A)	63
Figure 5-9 Comparison of computing times (Case B)	64
Figure 5-10 Comparison of computing times (Case C)	64
Figure 5-11 Comparison of computing times (Case D)	65
Figure 5-12 Comparison of computing times (SWM vs FEM)	66
Figure 5-13 Effects of different smoothing constants	67

List of Tables

Table 2-1 Term frequency in the sample case.....	9
Table 2-2 Calculation results of the sample case.....	10
Table 2-3 Algorithm for constructing a sub-graph	13
Table 3-1 Variables notation for the TFIDF algorithm	19
Table 3-2 Calculation results using the modified TFIDF algorithm.....	21
Table 3-3 Algorithm for constructing the node set in FEM	24
Table 3-4 Algorithm for constructing the node set in SWM.....	25
Table 3-5 RP_p value of the node in the simple graph	31
Table 3-6 Page value of the node in the simple graph.....	32
Table 3-7 Exponential smoothing algorithm.....	36
Table 4-1 Sample of byte stream returned by Yahoo	39
Table 4-2 HTML document structure.....	40
Table 4-3 Primary search scenario.....	41
Table 5-1 Query terms and search URLs.....	45
Table 5-2 Part of the URLs extracted without using the TFIDF algorithm.....	47
Table 5-3 The 10 URLs labeled in X-axis of Figure 5-3	48
Table 5-4 The 10 URLs labeled in X-axis of Figure 5-4	49
Table 5-5 Part of the searched URLs guided by the ES algorithm (Case A)	53
Table 5-6 Part of the searched URLs guided by the HITS algorithm (Case A)	54
Table 5-7 Part of the searched URLs guided by the ES algorithm (Case B)	56
Table 5-8 Part of the searched URLs guided by the HITS algorithm (Case B)	57
Table 5-9 Part of the searched URLs guided by the ES algorithm (Case C)	59
Table 5-10 Part of the searched URLs guided by the HITS algorithm (Case C)	60
Table 5-11 Part of the searched URLs guided by the ES algorithm(Case D)	61
Table 5-12 Part of the searched URL guided by the HITS algorithm(Case D).....	62
Table 5-13 Hub and authority values of the nodes in the two graphs.....	68

Table 5-14 Page values in the two graphs in Figures 5-14 and 5-15.....	69
Table 5-15 Page values after modification.....	70
Table 5-16 New search results guided by the ES algorithm (Case A).....	71
Table 5-17 New search results guided by the HITS algorithm (Case A).....	72
Table 5-18 New search results guided by the ES algorithm (Case B).....	73
Table 5-19 New search results guided by the HITS algorithm (Case B).....	74
Table 5-20 New search results guided by the ES algorithm (Case C).....	75
Table 5-21 New search results guided by the HITS algorithm (Case C).....	76
Table 5-22 New search results guided by the ES algorithm (Case D).....	77
Table 5-23 New search results guided by the HITS algorithm (Case D).....	78

List of Acronyms

DGM	Directed Graph Model
ES	Exponential Smoothing
EWMA	Exponential Weighted Moving Average
FEM	Free Expansion Model
HITS	Hyperlink Induced Topic Search
HTML	Hyper Text Mark-up Language
IR	Information Retrieval
SWM	Sliding Window Model
TFIDF	Term Frequency vs Inverted Document Frequency
URL	Universal Resource Locator
VSM	Vector Space Model
WWW	World Wide Web
W3C	World Wide Web Consortium

Nomenclature

a_p	The authority value of page p
A_k	Authority vector of iteration k , $A_k = (a_{1k}, a_{2k}, \dots, a_{pk})$
C_i	The set of new nodes in the i^{th} level search
CT	The total computing time
D	A document set
$\ D\ $	The mean norm of the document in a document collection
d_i	The i^{th} document in a document set
$\ d_i\ $	The norm of the i^{th} document in a document set
E	The set of arcs in a graph
$f(w)$	The frequency of a word in a document collection
$F_{(i,j)}$	The forecasting value of the j^{th} node in the i^{th} level search
G	The graph of WWW
h_p	The hub value of page p .
H_k	Hub vector of iteration k , $H_k = (h_{1k}, h_{2k}, \dots, h_{pk})$
m	The number of query terms
M	The adjacency matrix of a graph
N	The total number of documents in the collection
n_k	The number of documents that contain term k
$PV_{(i,j)}$	The page value of the j^{th} node in the i^{th} level search
Q	A query term set
$r(w)$	The rank of a word or term in a document collection
$RC(q, d_i)$	The degree of content relevancy between query q and document d_i
RP_p	The degree of importance of page p

$RV_{(i,j)}$	The normalized page value of the j^{th} node in the i^{th} level search
S_i	The node set in the i^{th} level search
$Sim(q, d_i)$	The similarity between query q and document d_i
T_F	The time of calculating forecasting value
T_G	The time of computing the hub value, the authority value
tf_{ik}	The frequency of term k that appears in the i^{th} document
TV	The total RV value of the searched pages in a certain search level
U_i	The sub set of C_i
V	The set of nodes in a graph
W_k	The weight of the k^{th} term in a query
w_{ik}	The weight of the k^{th} term in the i^{th} document
$x^{(p)}$	The authority value of a page in Kleinberg's iteration algorithm
$y^{(p)}$	The hub value of a page in Kleinberg's iteration algorithm
α	The smoothing constant
β	The weighting factor for calculating RP_p

Table of Contents

Abstract	i
Acknowledgements	ii
List of Figures	iii
List of Tables	iv
List of Acronyms	vi
Nomenclature	vii
Chapter 1 Introduction and Motivation.....	1
1.1 Background.....	1
1.2 Motivation and Objectives.....	2
1.3 The Organization of the Thesis.....	2
Chapter 2 Literature Review.....	4
2.1 Vector Space Model and Term Weighting Scheme.....	4
2.1.1 Vector Space Model (VSM).....	4
2.1.2 Zipf's Law and Term Weighting Scheme.....	4
2.2 Directed Graph Model.....	9
2.2.1 Directed Graph Model.....	10
2.2.2 Constructing a Directed Sub-graph of WWW.....	11
2.2.3 Computing Steady Hub Value and Authority Value.....	13
2.3 Final Remarks.....	17
Chapter 3 Proposed Dynamic Search and Distillation Approach.....	18
3.1 A Modified TFIDF Algorithm.....	18

3.2 Web Node Models	20
3.2.1 Free Expansion Model	22
3.2.2 Sliding Window Model.....	22
3.3 Page Value	26
3.4 Search Guidance	31
3.4.1 Forecasting Future Value for a Node in a Graph.....	32
3.4.2 Size of the Node Set in SWM Using ES algorithm.....	34
3.5 Overall Algorithm of the Search Agent	36
Chapter 4 Implementation.....	38
4.1 Search Depth.....	38
4.2 Obtaining the Seed URLs	38
4.3 Scanning Pages	38
4.4 Primary Search Scenario.....	40
4.5 Class Diagram.....	41
Chapter 5 Results and Analysis	45
5.1 Effects of Content-relevancy Value.....	45
5.2 Search Results and Comparisons.....	50
5.2.1 Search Effectiveness Metrics	50
5.2.2 Comparisons between the ES and Original HITS.....	51
5.2.3 Computing Time Comparison between FEM and SWM.....	65
5.2.4 Effects of Different Smoothing Constants.....	66
5.2.5 Results Using Modified Page Value Metrics.....	67
Chapter 6 Conclusion and Future work	79
6.1 Conclusion	79
6.2 Future Work.....	79
References.....	81
Appendices.....	87
Appendix 1 Source Code.....	87
Appendix 2 The Searched URLs.....	142
Appendix 3 The New Results.....	169

Chapter 1 Introduction and Motivation

1.1 Background

In the last decades, the rapid increase of computer systems that are inter-connected in on-line networks has resulted in a considerable increase in the amount of information available on-line (Moukas, 1996, Moukas and Maes, 1998). World-Wide-Web (WWW) has become a global community that represents a rich hypertext collection (Kleinberg, 1999). With the increasing needs of utilizing the Internet resource, efficient Information Retrieval (IR) from Internet has become a major concern and different models have been devised in the last few years to retrieve the information, such as the Vector Space Model (VSM) and the Directed Graph Model (DGM). Many search engines and agents integrating these models and algorithms have emerged (www.altavista.com, www.copernic.com, www.google.com, www.searchenginewatch.com, www.yahoo.com). Some have gained great success, such as Google. But it appears that few of them, if any, could relieve the user from a tedious search process which is still imposing a heavy burden on the user due to the time required for searching the right sites, browsing the obtained information, filtering out the irrelevant contents, and organizing the useful and valuable information for future use (Yuhono and Lee, 1996, Lam, 2001). Besides, the gathered information through a search engine may or may not be relevant to the user's intention (Flake et al., 2000, Glover et al., 2001). It is therefore necessary to develop a web information retrieval and distillation agent to assist the user in retrieving and selecting the useful resources on the Internet without much human effort. Such a search process can also assist us in building our own database on reference relationships among different web pages for a specific query. Though different search agents aiming to assist the user's search had been designed (Balabanovic and Shoham, 1995, Yuhono and Lee, 1996, Brin and Page, 1998, Chen and Sycara, 1998, Dean and Henzinger, 1999), each search agent has its own deficiency and limitations. More specifically, these search agents still need human interaction and supervision and can not completely relieve the

user from a time-consuming search process. Also there is a lack of systematic approach to quantify the value of the information for the user's needs.

1.2 Motivation and Objectives

This thesis stems from the following observations: (1) There is a lack of an approach which utilizes both link and content information. The link-based approach, such as HITS or PageRank used in Google, is efficient in terms of search time, but falls short in information quality. On the other hand, the content-based approach tends to be inefficient. The integration of the two should compensate for drawbacks of both. (2) There is a need to devise a more comprehensive criterion to measure the value or importance of a web page. The previous criteria such as hub-authority values and similarity are inadequate and often mis-lead the search process. (3) The current search engines or agents tend to be static in the search process, i.e., they are focused on a given set of pages of a database. Any additional in-depth search will involve human effort.

Motivated by the above observations, this thesis is directed towards: a) developing a combined link and content based search approach which considers both the link information and content information to improve information quality; b) accordingly proposing a more comprehensive criterion to measure the value of a web page by considering both link information and content information; c) applying the exponential smoothing method to guide the search process and improve search efficiency by forecasting or predicting more valuable search paths and discarding less valuable paths; and d) constructing a sliding window model to facilitate multi-level in-depth dynamic search process. Then a search agent will be designed based on all the above components.

1.3 The Organization of the Thesis

This thesis is organized as follows: Chapter 2 reviews the basic algorithms and models in IR such as the VSM and DGM. Their drawbacks and limitations will also be discussed.

In Chapter 3, the integrated link-content approach will be developed, the new page value metric will be proposed, the ES based guiding algorithm will be constructed and the sliding window model will be suggested. The new search agent will be developed incorporating all these elements. In Chapter 4, we describe the implementation of the search agent. In Chapter 5, we conduct a set of experiments to test the performance of the proposed algorithms and models. Chapter 6 concludes the thesis. Future work will also be discussed in this chapter.

For the purpose of current discussion, a URL, a page or a link is treated as a node in proposed web node model. A query term is typically a single word. However it can consist of several words that are commonly used together.

Chapter 2 Literature Review

In this chapter, we will review the basic principles and models involved in information retrieval (IR). The drawbacks and limitations of these models and principles will also be discussed.

2.1 Vector Space Model and Term Weighting Scheme

2.1.1 Vector Space Model (VSM)

The VSM has been widely used in traditional IR (Lewis, 1996, <http://isp.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html>). This model creates a space in which both documents and queries are represented by vectors. The vector space has as many dimensions as the number of terms in the query (Figure 2-1). Each dimension corresponds to a single query term. For a fixed collection of documents, an m - dimensional vector is generated for each document where m is the number of unique terms in the query. Then, a vector similarity function can be used to compute the similarity between a document and a query. The similarity in VSM is determined by using associative coefficients based on the inner product of the document vector and query vector, where word overlap indicates similarity. The inner product is usually normalized. The most popular similarity measure is the cosine coefficient, which measures the angle between the document vector and the query vector (Figure 2-1) (<http://www.cs.cornell.edu/courses/cs430/2001fa/sylabus.html>). Most search engines use similarity measures based on this model to rank web documents (<http://isp.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html>). The notion of similarity had also been seen in reference (Hersovici et al., 1998).

2.1.2 Zipf's Law and Term Weighting Scheme

2.1.2.1 Zipf's Law

Obviously, each element in a document vector or a query vector must be assigned a real value to compute the similarity in vector space model. However, there had been many versions of assigning the values to the elements in the vector. For instance, some assigned the values of the elements simply in terms of its occurrence frequency in the document (<http://www.cs.cornell.edu/courses/cs430/2001fa/syllabus.html>). When some IR experts took Zipf's law into consideration later, they realized that the above treatment to the elements of document vector may result in a great deviation and it should be improved.

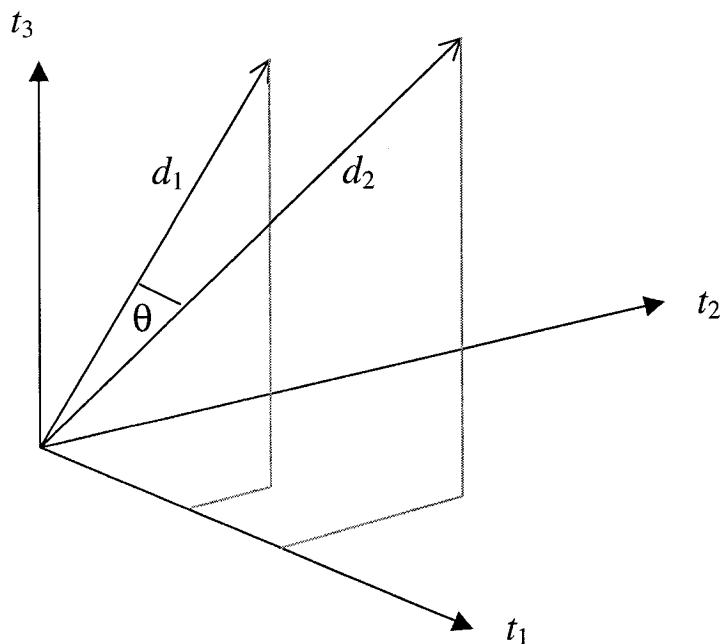


Figure 2-1 Vector space model

d_1 and d_2 represent two documents respectively. t_1 , t_2 and t_3 represent three query terms. θ is the angle between d_1 and d_2 in this space. The cosine value of angle θ measures the similarity between document d_1 and d_2 .

Zipf's law (Salton, 1988), named after the Harvard linguistic professor George Kingsley Zipf, states:

If a word, w , in a collection is ranked by its frequency, the rank and the frequency of the word roughly fit the relation (Figure 2-2):

$$r(w)f(w) = c \quad (2.1)$$

where c is a constant, $r(w)$ the rank of a word, and $f(w)$ the frequency of a word. $r(w)$ is determined by the word's frequency. The higher the term frequency, the higher the term rank will be. The smaller the number, the higher the rank.

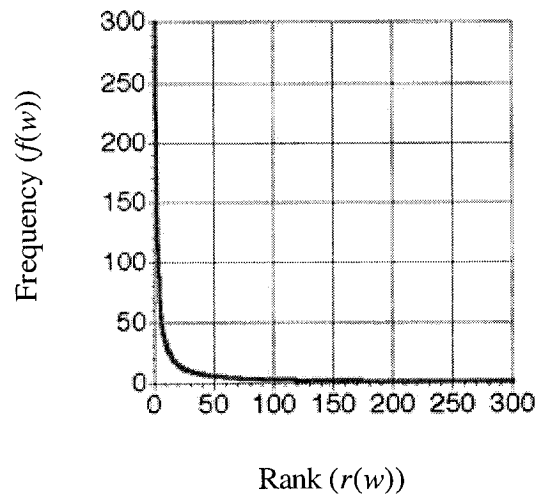


Figure 2-2 Zipf's law

This suggests that some terms are more effective than others in retrieval. Moreover, some terms may occur with a substantially high frequency in some documents while having relatively overall low collection frequency. Hence, Salton (1989) proposed the Term Frequency vs Inverted Document Frequency (TFIDF) weighting scheme, which is an algorithm used to weight the word in a document collection, to quantify these concepts and to control the specificity when computing the similarity.

2.1.2.2 TFIDF Weighting Scheme

This scheme, as its name suggests, deals with the term frequency and the document frequency. In the Term Weighting vs Inverted Document Frequency scheme,

- (a) Weight is proportional to the number of times that the term appears in the document.
- (b) Weight is inversely proportional to the number of documents that contain the term.

The weight of a term is given by (http://www.sims.berkeley.edu/courses/is240/s02/Lectures/Lecture7_240.ppt):

$$w_{ik} = \frac{tf_{ik} \log(N/n_k)}{\sqrt{\sum_{j=1}^m tf_{ij}^2 [\log N/n_j]^2}} \quad (2.2)$$

where w_{ik} is the weight of the k^{th} term in the i^{th} document, N is the total number of documents in the collection, tf_{ij} is the frequency of term j that appears in the i^{th} document, n_j is the number of documents that contain term j , m is the number of query terms and $\log(N/n_j)$ is the Inverted Document Frequency (IDF) of term j .

The similarity between the i^{th} document and the j^{th} document is given by (http://www.sims.berkeley.edu/courses/is240/s02/Lectures/Lecture7_240.ppt):

$$Sim(d_i, d_j) = \sum_{k=1}^m w_{ik} w_{jk} \quad (2.3)$$

where d_i and d_j are the i^{th} and j^{th} documents respectively and m is the number of query terms.

To retrieve the documents that match a particular query, the query can be treated as a very small unstructured document, and the weight of the term in the query is given by the following equation (http://www.sims.berkeley.edu/courses/is240/s02/Lectures/Lecture7_240.ppt):

$$W_k = \frac{tf_k}{\sqrt{\sum_{j=1}^m tf_j^2}} \quad (2.4)$$

where tf_k is the occurrence frequency of the k^{th} term in a query , m is the number of query terms.

The similarity between a query and a document is given by:

$$Sim(q, d_i) = \sum_{k=1}^m W_k w_{ik} \quad (2.5)$$

However, Equation 2.2 is not a standard one because there may exist the case where the number of documents that contain term k is equal to the number of total documents in a collection. Thus, the IDF component of Equation 2.2 will be zero. Consequently, it will make the weight of the k^{th} term in a document zero. To avoid such a difficulty, Jones (2001) proposed a general term weighting scheme. In his scheme, the IDF is given by:

$$IDF = \log(N / n_k) + 1$$

and the standard term weighting equation becomes:

$$w_{ik} = \frac{tf_{ik} [\log(N / n_k) + 1]}{\sqrt{\sum_{j=1}^m tf_{ij}^2 [\log(N / n_j) + 1]^2}} \quad (2.6)$$

Therefore, Equation 2.6 is used to assign the weight for a term instead of Equation 2.2.

By calculating the similarity between the query and the given set of documents, those documents obtaining high values of similarity are retrieved in priority.

This weighting scheme had been used in some agents. For example, Chen and Sycara (1998) designed a personal search agent implementing this weighting scheme to assist the user in organizing the results.

2.1.2.3 Drawbacks of TFIDF Algorithm

The TFIDF weighting scheme has some significant drawbacks, which can be illustrated as follows. Suppose the query terms are “java developer”, and we are given two documents, d_1 and d_2 , of equal size. The occurrence frequencies of the terms in the two documents are listed in the following table:

	Java	Developer
d_1	2	2
d_2	4	3

Table 2-1 Term frequency in the sample case

Based on Equation 2.6, the calculation results are listed in Table 2-2.

As shown in Table 2-2, the similarity between the query and document d_2 is lower than that between the query and document d_1 . However, Table 2-1, clearly shows both query terms in document d_2 appear more frequently. As a result, the information retrieval based on such a weighting scheme may mis-lead the search process.

2.2 Directed Graph Model

Clearly, with TFIDF, documents are retrieved and ranked depending on how well they match a specific query (Jones, 2001). While there may exist a large volume of documents on the Web that match a particular query, the user may retrieve a large amount of pages

Calculation Results		
Symbol	Notation	Value
n_1	The number of documents containing “java ”	2
n_2	The number of documents containing “developer”	2
m	The number of query terms.	2
tf_{11}	The frequency of “java” in d_1	2
tf_{21}	The frequency of “java” in d_2	4
tf_{12}	The frequency of “developer ” in d_1	2
tf_{22}	The frequency of “developer” in d_2	3
N	The total number of documents in our collection	2
w_{11}	The weight of term “java” in d_1	0.707
w_{21}	The weight of term “java” in d_2	0.8
w_{12}	The weight of term “developer” in d_1	0.707
w_{22}	The weight of term “developer” in d_2	0.6
W_1	The weight of term “java” in the query	0.707
W_2	The weight of term “developer” in the query	0.707
$Sim(q, d_1)$	The similarity between the query and document d_1	1.0
$Sim(q, d_2)$	The similarity between the query and document d_2	0.9898

Table 2-2 Calculation results of the sample case

that are not authoritative under this condition. Thus, the directed graph model was proposed to measure the importance of the document (Kleinberg, 1999, Deo and Gupta, 2001, Kleinberg and Lawrence, 2001).

2.2.1 Directed Graph Model

It is understood that WWW is an intricate community that represents hyper-linked text (Kleinberg and Lawrence, 2001), and most of the web pages are multi-referenced (Kleinberg, 1999). Therefore, the reference relationship of web pages is very important in information retrieval.

To determine the reference relationship of the web pages in WWW, Kleinberg (1999) introduced a Directed Graph Model (DGM). According to Kleinberg's analysis (1999), "hyperlinks encode a considerable amount of latent human judgment, the creation of a link on the WWW represents a concrete indication of the following type of judgment: the creator of page P, by including a link to page q, has in some measure conferred authority on q".

This model is based on the relationship that exists between the authorities for a topic and those pages that link to many related authorities. The page of the latter type is referred to as a *hub*. The objective of this model is to find the most authoritative page in a given collection.

2.2.2 Constructing a Directed Sub-graph of WWW

In this section, we describe a directed sub-graph of WWW and the algorithm to construct a directed sub-graph.

2.2.2.1 The Directed Graph

A directed graph consists of a set of *nodes*, denoted by V and a set of *arcs*, denoted by E . Each arc is an ordered pair of nodes (u, v) representing a directed connection from u to v . The *out-degree* of a node u is the number of distinct arcs $(u, v_1) \dots (u, v_k)$ (the number of links from u), and the *in-degree* is the number of distinct arcs $(v_1, u) \dots (v_k, u)$ (the number of links to u). A path from node u to node v is a sequence of arcs $(u, u_1), (u_1, u_2), \dots (u_k, v)$. One can follow such a sequence of arcs to "walk" through the graph from u to v . Note that a path from u to v does not imply a path from v to u (Broder et al., 2000).

2.2.2.2 Algorithm to Construct a Sub-graph

To find the most authoritative page in a subset of WWW, a directed graph must be constructed to determine the multi-reference relationships among the pages in the subset. We can view any collection V of hyperlinked pages as a directed graph $G = (V, E)$. The nodes correspond to the pages, and a directed edge $E(p, q)$ indicates the presence of a link from p to q . From a graph G , we can isolate small regions, or *sub-graphs*, in the following way (Kleinberg, 1999). First, we collect t pages from a certain search engine. These t pages will be referred as the root set R . Second, try to find all the pages that the pages in the root link to and the pages that link to all the pages in the root. Third, add all these pages to the root set to form the base set. Table 2-3 describes the algorithm for constructing a sub-graph and Figure 2-3 shows the graphic representation of a sub-graph.

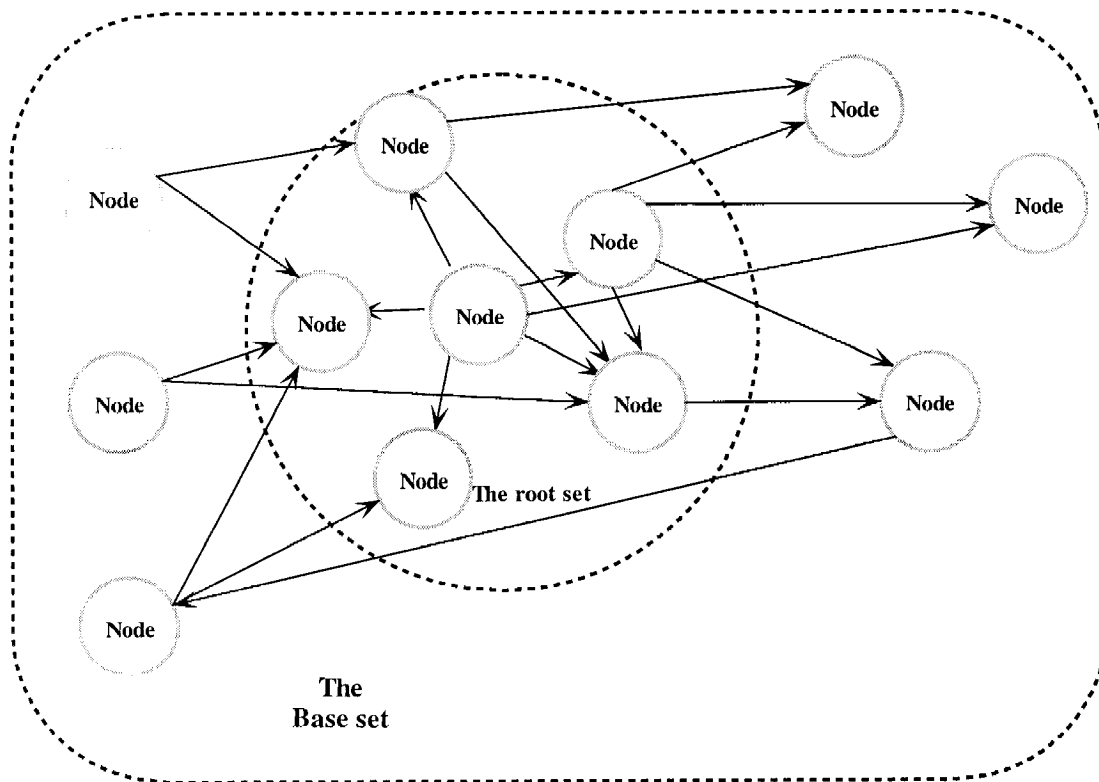


Figure 2-3 Sub-graph of a directed graph model

Sub-graph (Q, E, t, d)

Q : A query string.

E : a search engine.

t, d : natural numbers.

Let R_Q denote the top t results of E on Q .

Set $S_Q := R_Q$

For each page $p \in R_Q$

Let $R^+(p)$ denote the set of all pages that p points to.

Let $R^-(p)$ denote the set of all pages that point to p .

Add all pages in $R^+(p)$ to S_Q .

Let $|R^-(p)|$ denote the number of pages pointing to p .

If $|R^-(p)| < d$ then

Add all pages in $R^-(p)$ to S_Q .

Else

Add an arbitrary set of d pages from $R^-(p)$ to S_Q

Return S_Q .

Table 2-3 Algorithm for constructing a sub-graph

2.2.3 Computing Steady Hub Value and Authority Value

In this section, we describe how to compute the hub value and the authority value mentioned in Section 2.2.1.

2.2.3.1 Hubs and Authorities

As defined in Section 2.2.1, a hub is a node pointing to other pages, while an authority is a node pointed by other pages. In a graph, each node can be both a hub and an authority.

The graphic representations of the hub page and the authority page clearly describe these relationships (Figure 2-4 and Figure 2-5).

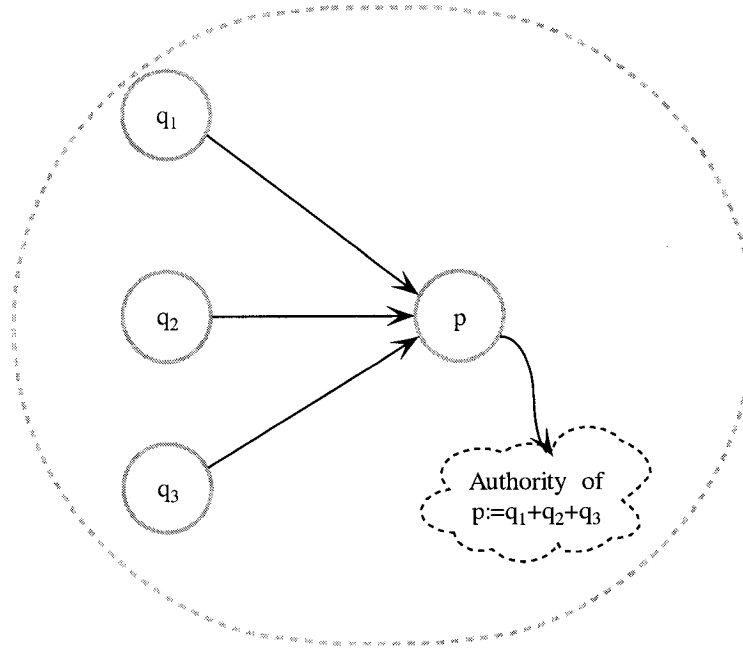


Figure 2-4 Authority

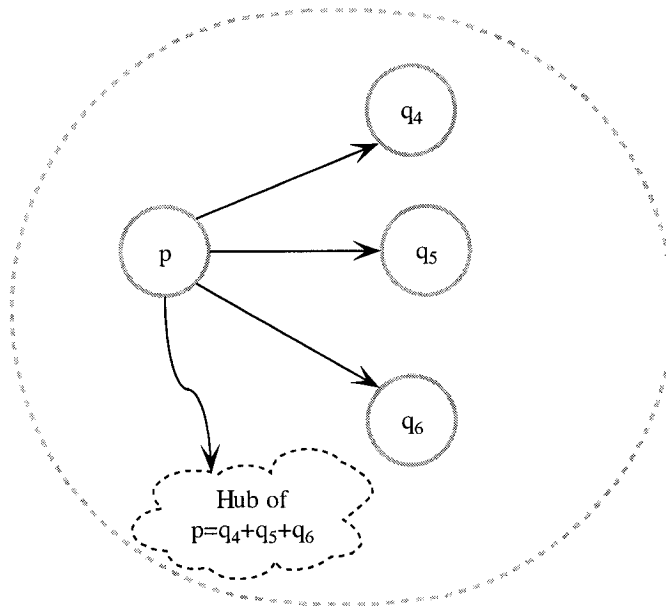


Figure 2-5 Hub

2.2.3.2 Iterative Algorithm

Kleinberg (1999) employed an iterative algorithm to compute the hub and authority values. Since this algorithm will be used in the following chapters, it is briefly described as follows.

For any given page (also referred to as a node) P , let $x^{(p)}$ denote the authority value of page P and $y^{(p)}$ denote the hub value. For any page $q, q \in E(p, q)$, the algorithm updates the authority and hub values of each page as follows:

$$x^{(p)} \leftarrow \sum_{q:(q,p) \in E} y^{(q)}, \text{ we call it } I \text{ operation.}$$

$$y^{(p)} \leftarrow \sum_{q:(p,q) \in E} x^{(q)}, \text{ we call it } O \text{ operation}$$

where $q:(p,q) \in E$ is a page q that links to page p in the directed edge set E . Thus I and O are the basic means by which hubs and authorities reinforce one another. Figure 2-6 shows the flow chart of this algorithm.

Kleinberg (1999) has proven that such a sequence of iterations, when normalized, converges to the principal eigenvector of matrix $M^T M$, where M is an adjacency matrix whose entry (i, j) is 1 if there exists a link between the i^{th} page and the j^{th} page, or 0 otherwise.

Kleinberg's experiments (1999) show that the convergence of the algorithm is quite rapid. Normally the hub and authority values become stabilized after 20 to 40 iterations. The algorithm used to compute the hub and authority values of a node in a directed graph is called the HITS algorithm (Hyperlink Induced Topic Search). This algorithm has been widely used in finding the most related pages on the Web (Balabanovic and Shoham, 1995, Dean and Henzinger, 1999). It suggests that a page's importance can be determined in terms of its reference relationship, or hub and authority values, in a graph.

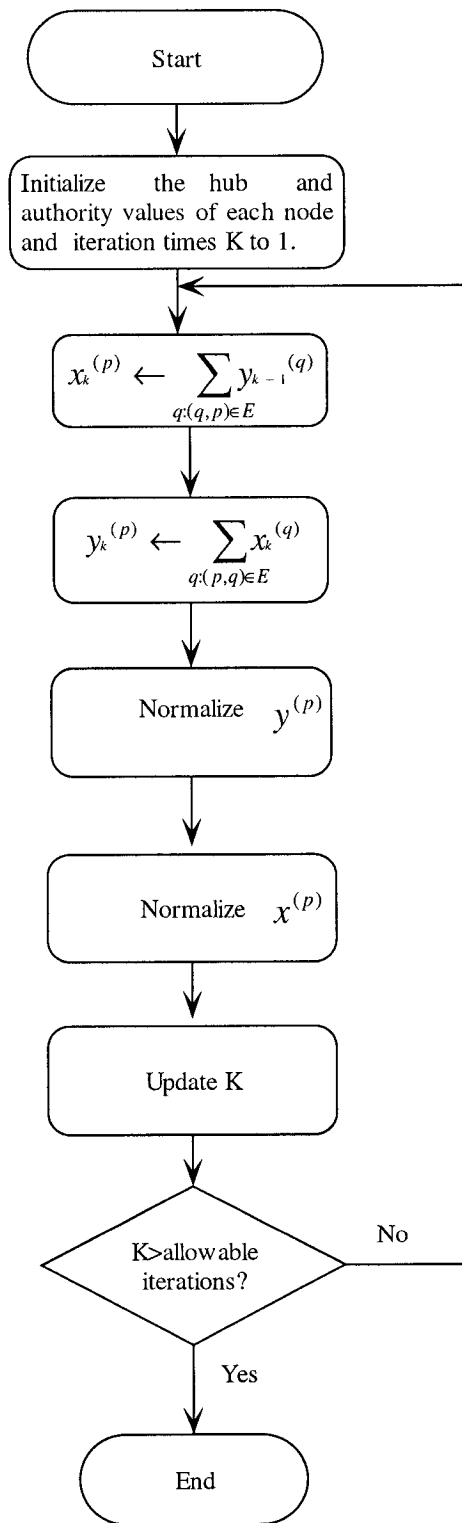


Figure 2-6 Flow chart of iterative algorithm

2.3 Final Remarks

The above models and principles provide prototypes for the search engine to rank and present search results to the user. However, these models and algorithms are mostly applied to a static database or a pre-specified sub-set of WWW for ranking and retrieval. The quantity and the relevance of the information obtained using these models or algorithms apparently depend on the selection of such a sub-set of WWW. The specification of the sub-set and its size is quite subjective. The selection of a large sub-set will be likely to improve the search result, but the computing time and search time will increase drastically. Our purpose is however to design an information and distillation agent that is to be used to, if necessary, crawl the Web continuously. The scope and the direction of the search can not be pre-determined since otherwise the entire web has to be searched which is obviously infeasible. This also makes it impossible to adopt the static approaches reviewed above. For this reason, a dynamic search approach that is self-guided in determining the search direction is proposed. The details of this approach are presented in the next chapter.

Chapter 3 Proposed Dynamic Search and Distillation Approach

As reviewed in Chapter 2, the TFIDF approach has serious drawbacks in retrieving documents based on similarity. Besides, although the directed graph model provides an important heuristic and measure to evaluate the importance of the documents, it is only applied to a static set of documents. In our case, however the amount and the source of the data can not be pre-determined and vary with time. A dynamic approach is needed. Also the TFIDF approach does not consider link information whereas the HITS method disregards the content information. This indicates a need for an integrated approach incorporating both link and content information. For these reasons, in this chapter we develop: a) a modified TFIDF algorithm to prevent the drawbacks present in the original TFIDF algorithm; b) a sliding window model for dynamic search; c) a new page value measure incorporating both link and content information; and d) an ES based forecasting method to guide the dynamic search process.

3.1 A Modified TFIDF Algorithm

To avoid the problem present in the TFIDF weighting scheme as indicated in Chapter 2 and illustrated in Table 2-2, we defined a new performance indicator,

$$RC(q, d_i) = Sim(q, d_i) \frac{\|d_i\|}{\|D\|} \quad (3.1)$$

where $Sim(q, d_i)$ is the similarity between the query q and document d_i , $\|d_i\|$ is the norm of document i , and $\|D\|$ is the mean norm of a document set.

All the parameters and a few other parameters to be used later are summarized and explained in Table 3-1.

Equation 3.1 shows that both similarity and the occurrence frequency concerns are incorporated into the new indicator RC . The procedure of computing the RC value is given in Figure 3-1. The modified TFIDF algorithm will be used in Section 3.3 to calculate page value PV , a composite indicator of information relevance and importance.

Definition	
d_i	A document, labelled i .
D	An n -tuple set of documents, $D = \{d_1, d_2, d_3 \dots d_n\}$
t_j	A query term, labelled j .
Q	An m -tuple set of terms, $Q = \{t_1, t_2, t_3 \dots t_m\}$.
m	Number of terms in set Q .
N	Total number of documents.
n_j	Occurrence frequency of the documents in set D that contains term t_j .
tf_{ij}	Occurrence frequency of term t_j in document d_i .
W_j	Query weight of term t_j .
w_{ij}	The TFIDF weight of term t_j in document d_i .
$\ d_i\ $	The Norm of document d_i , $\ d_i\ = \sqrt{\sum_{j=1}^m tf_{ij}^2}$.
$\ D\ $	The Mean norm of set D , $\ D\ = \frac{\sum_{i=1}^n \ d_i\ }{n}$.
$Sim(q, d_i)$	The similarity between query q and document d_i .
$RC(q, d_i)$	The content-relevancy level between query q and document d_i .

Table 3-1 Variables notation for the TFIDF algorithm

To test the modified term weighting scheme, we apply it to the same case discussed in Chapter 2. This time, we retrieve the documents based on RC value instead of similarity. Table 3-2 listed the calculation results of the RC values of documents d_1 and d_2 .

Calculation results	
Variable	Value
$\ d_1\ $	2.828
$\ d_2\ $	5
$\ D\ $	3.914
$RC(q, d_1)$	0.723
$RC(q, d_2)$	1.264

Table 3-2 Calculation results using the modified TFIDF algorithm

Obviously, document d_2 has a higher RC value now and the results better reflect our intention. Therefore, the RC value provides a more reliable measure for prioritizing documents in terms of both similarity and term occurrence frequency. The flowchart of this algorithm is shown in Figure 3-1.

3.2 Web Node Models

As reviewed in Chapter 2, the HITS (Hyperlink Induced Topic Search) algorithm is only applicable to static and small node sets. The quality of the retrieved information is therefore greatly dependent on the selection of a relevant and static node set, which itself is a challenging task. To alleviate this difficulty, a Sliding Window Model (SWM) is proposed in this study. For comparison purpose, the Free Expansion Model (FEM) is also presented in this section.

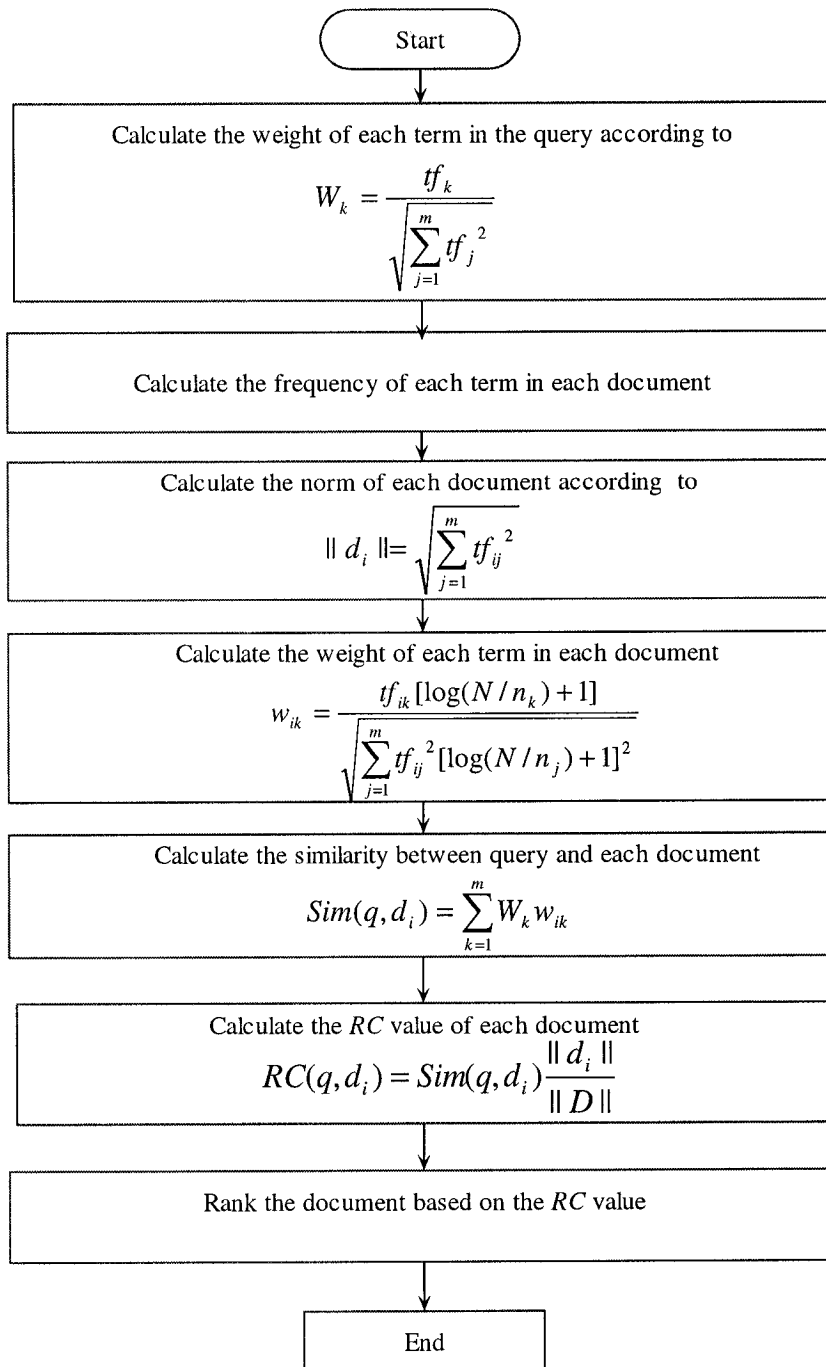


Figure 3-1 Flow chart of computing the RC value

3.2.1 Free Expansion Model

The FEM, as its name suggests, allows the size of the node set to expand without any control. It works as follows:

Let i represent the search level and let S_i denote the set that the graph relies on in the i^{th} level of search, S_i will be the union of all the previous nodes:

$$S_i = C_1 \cup C_2 \cup \dots \cup C_i \quad (3.5)$$

where C_i is the set of new nodes found in the i^{th} level search. Alternatively,

$$S_i = S_{i-1} \cup C_i . \quad (3.6)$$

This model is graphically represented in Figure 3-2. In Figure 3-2, the number in the center of each node denotes the search level. Clearly, the significant drawback of this model is the increased computing and the search time as the size of the node set becomes large. Table 3-3 describes the algorithm for constructing the node set using FEM.

3.2.2 Sliding Window Model

To control the size of the node set for each next search level, we develop a Sliding Window Model (SWM) in this section.

The SWM permits new nodes to be added into the current node set, and discards the nodes obtained a certain search levels ago. The total node collection for any search level i is determined by

$$S_i = C_i \cup C_{i-1} \cup C_{i-2} \cup \dots \cup C_{i-N_w+1} \quad (3.7)$$

where N_w is the number of search levels contained in the current window.

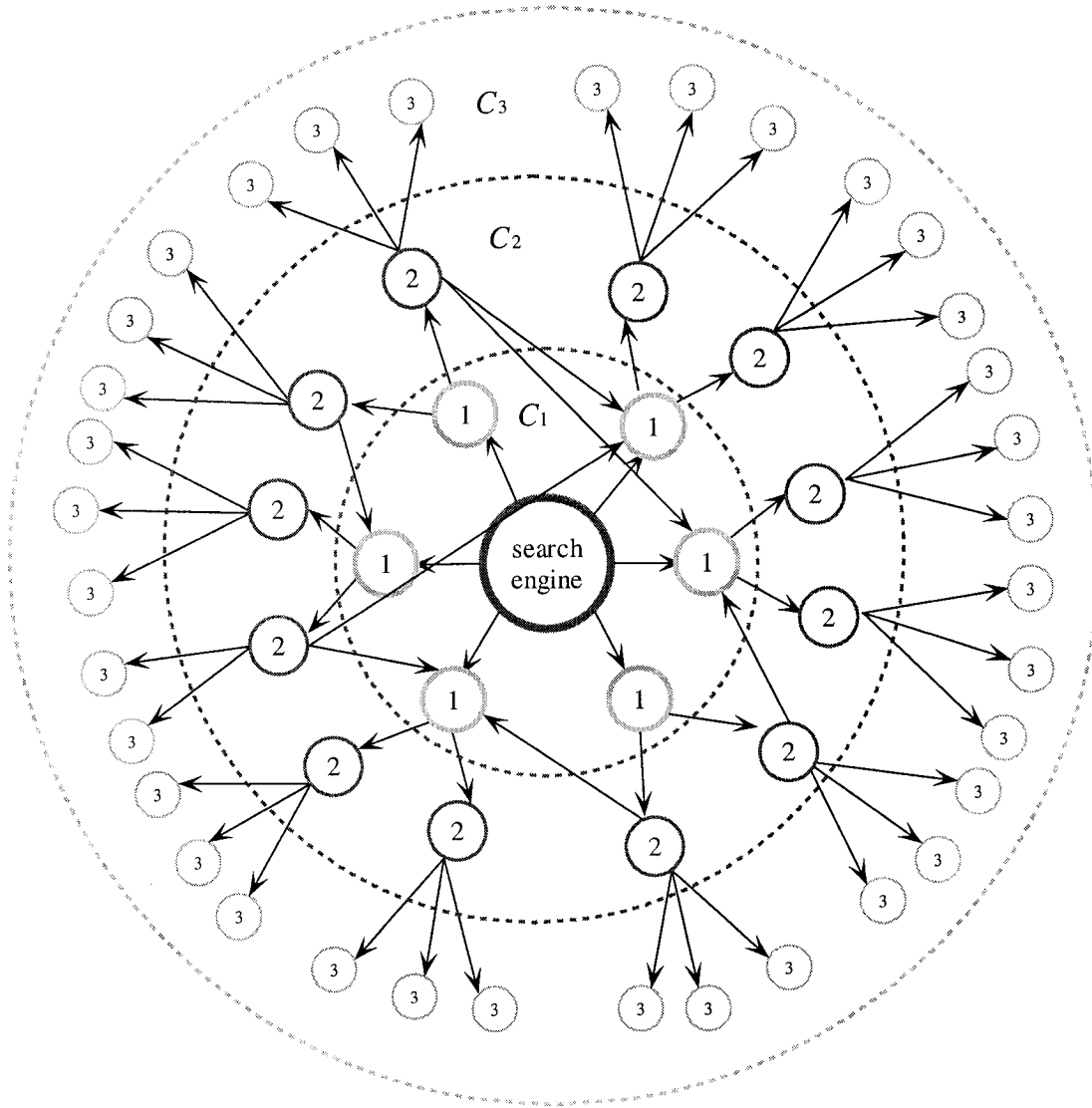


Figure 3-2 Free Expansion Model

Let Q denote the query string, Y the search engine (e.g., Yahoo),
 R_q the URL 's obtained using Y on Q , S_i the node set of the i^{th} level search.
Set $i=1$, $S_i = C_i = R_q$;
For each page $p \in C_{i-1}$,
Let C_i denote the set of all new pages pointed by any page in C_{i-1} ;
 $S_i := S_{i-1} \cup C_i$;
Return S_i and update i .
End.

Table 3-3 Algorithm for constructing the node set in FEM

When the search proceeds to next level, the oldest nodes will be deleted and new nodes will be added. Hence, the node collection for the next search level is

$$S_{i+1} = C_{i+1} \cup C_i \cup C_{i-1} \cup \dots \cup C_{i-N_W+2} \quad (3.8)$$

Figure 3-3 provides a graphical presentation of the proposed sliding window model. In Figure 3-3, N_W is 3. The algorithm for constructing node sets using this model is given in Table 3-4. If we focus on the top m nodes of each search level, then, the size of the node set in each search level is now limited to $N_W \times m \times n$, assuming each page contains n links.

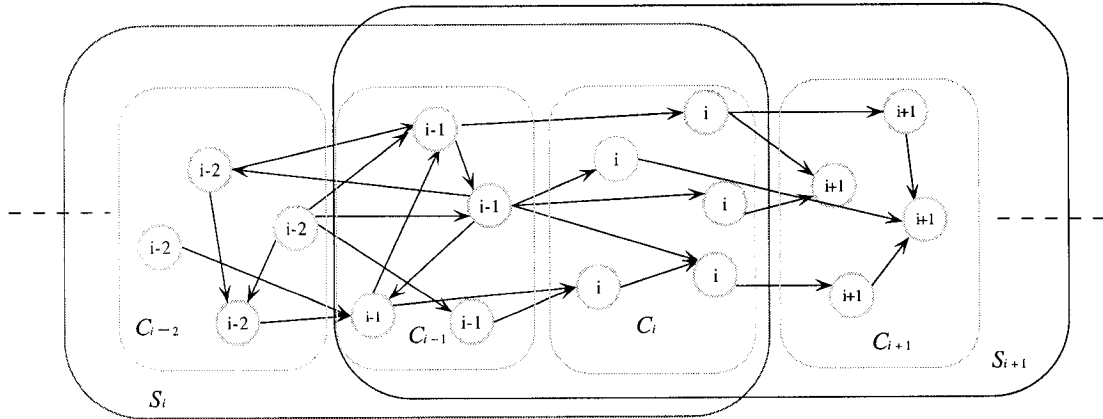


Figure 3-3 Sliding Window Model

Let Q denote the query string, Y the search engine (e.g., Yahoo),
 R_q the URL's obtained using Y on Q , i the i^{th} search level, S_i the node set of
the i^{th} search level.

Set $i=1$ and $S_i = C_i = R_q$.

For each page $p \in S_i$,

Let C_i denote the set of all new pages pointed by any page in C_{i-1} ;

If $i > N_w + 1$

$$S_i = C_i \cup C_{i-1} \cup C_{i-2} \cup \dots \cup C_{i-N_w+1};$$

Else

$$S_i = S_{i-1} \cup C_i$$

Return S_i and update i .

End.

Table 3-4 Algorithm for constructing the node set in SWM

3.3 Page Value

We have discussed that the VSM is used to measure the degree of content-relevancy of a page, and the DGM is used to measure the degree of importance of a page which may be reflected by the combined hub and authority value. Also, we have modified the TFIDF algorithm to make it more rational and precise in measuring the content-relevancy by RC as defined in Equation 3.1, and created the SWM for the purpose of maintaining a sustainable search process. In reality, the value of a page should be measured by both the content-relevancy and the degree of importance. For this reason, we define a composite indicator as follows:

$$PV_p = RC_p \times RP_p \quad (3.9)$$

where PV_p is the composite value of page p , RC_p the content-relevance level of page p , and RP_p the importance level of page p . RP_p can be calculated based on the hub and authority values as follows:

$$RP_p = \beta h_p + (1 - \beta) a_p \quad (3.10)$$

where h_p is the hub value, a_p the authority value and β the weighting factor between 0~1. Both h_p and a_p can be obtained using Kleinberg's algorithm as reviewed in Figure 2-6 (with a_p replacing x_k and h_p replacing y_k respectively). The main idea lies in the fact that a heavily visited page will have higher authority level and hence reflects the level of recognition. A page with a higher hub value contains a larger number of out-links, which may indicate indirect important information. Hence, the RP value contains both hub and authority values. However, since the more a page is cited, the more important this page is (Charkrabarti et al., 1999, Charkrabarti, 2001), the degree of importance of a page is mainly determined by its authority and more weight should be

assigned to a_p in the above equation. Also, the above equation can avoid unfairly assigning more weight to a page which has relatively higher hub and authority values.

To compare different pages obtained in each search level, we further define a normalized value of a page, RV_p , as given by

$$RV_p = \frac{PV_p}{\sum_{j=1}^N PV_j}, \quad (3.11)$$

where N is the total number of nodes or pages of a given set in a certain search level.

We use a simple graph (Figure 3-4) to illustrate the computation of RV . The procedure is illustrated as follows:

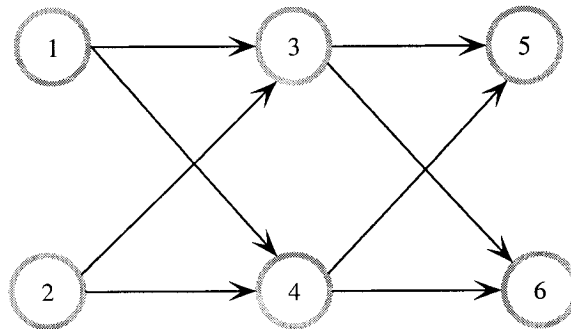


Figure 3-4 A simple graph

- **Calculating hub and authority values**

According to Kleinberg's iterative algorithm reviewed in Section 2.2.3.2, the adjacency matrices M and M^T for the graph in Figure 3-4 are:

$$M = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad M^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

Let A_k denote the authority vector and H_k the hub vector in the k^{th} iteration respectively.

For $k=0$, A_0 and H_0 are specified as:

$$A_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad H_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

when $k=1$,

$$A_1 = M^T H_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}$$

$$H_1 = MA_0 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$

A_1 and H_1 are normalized as follows:

$$A_1 := A_1 / \|A_1\| = \begin{pmatrix} 0 \\ 0 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} \quad H_1 := H_1 / \|H_1\| = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0 \\ 0 \end{pmatrix}$$

$$\text{where } \|A_1\| = \sqrt{0^2 + 0^2 + 2^2 + 2^2 + 2^2 + 2^2} = 4,$$

$$\|H_1\| = \sqrt{0^2 + 0^2 + 2^2 + 2^2 + 2^2 + 2^2} = 4.$$

when $k=2$,

$$A_2 = M^T H_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$H_2 = MA_1 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

A_2 and H_2 are normalized as follows:

$$A_2 := A_2 / \|A_2\| = \begin{pmatrix} 0 \\ 0 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} \quad H_2 := H_2 / \|H_2\| = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0 \\ 0 \end{pmatrix}$$

where $\|A_2\| = \sqrt{0^2 + 0^2 + 1^2 + 1^2 + 1^2 + 1^2} = 2$,

$$\|H_2\| = \sqrt{0^2 + 0^2 + 1^2 + 1^2 + 1^2 + 1^2} = 2.$$

Further iterations will not change the vectors and thus the final authority and hub vectors are the same as A_2 and H_2 . Each element in A_2 and H_2 is the corresponding a_p and h_p of a node in the graph.

- **Calculating RP_p**

Associated RP_p values can be calculated using Equation 3.10 based on the above a_p and h_p values. Assuming β is 0.1, the a_p , h_p and RP_p values for the six nodes in Figure 3-4 are summarized in Table 3-5.

Node	a_p	h_p	RP_p
1	0	0.5	0.05
2	0	0.5	0.05
3	0.5	0.5	0.5
4	0.5	0.5	0.5
5	0.5	0	0.45
6	0.5	0	0.45

Table 3-5 RP_p values of the nodes in the simple graph

- **Calculating PV_p**

Suppose each node has been searched and obtained a RC_p value defined in Equation 3.1. According to Equation 3.9, the RP_p , RC_p , PV_p and RV_p of each node in the graph are listed in Table 3-9.

Node	RC_p	RP_p	PV_p	RV_p
1	0.6	0.624	0.372	0.0507
2	0.8	0.624	0.496	0.0676
3	1.4	1.66	2.324	0.3166
4	1.2	1.66	1.992	0.2713
5	0.8	1.54	1.232	0.1678
6	0.6	1.54	0.924	0.1258

Table 3-6 Page values of the nodes in the simple graph

3.4 Search Guidance

To enhance the reasoning ability of the search agent, we employ the Exponential Smoothing (ES) method to guide the search. The idea is to use ES to predict the future

value of a node in a certain search level, and search the most promising nodes based on their forecasting values. Besides, the ES can help the search agent avoid collecting the potentially less valuable nodes, and thus can further control the size of the node set in each search level.

3.4.1 Forecasting Future Value for a Node in a Graph

The Exponential Smoothing (ES) is a very popular scheme used in the industrial forecasting. It assigns exponentially decreasing weights as the observations or data get older. The basic equation of exponential smoothing is defined as follows (Nahiam, 2001):

$$S_t = \alpha y_{t-1} + (1 - \alpha)S_{t-1}$$

where S_i stands for the smoothed observation or Exponential Weighted Moving Average (EWMA), and y stands for the original observation. The subscripts refer to the time periods, 1, 2, ..., n . Parameter α is called the smoothing constant, $0 < \alpha \leq 1$.

For the same token, in the context of web search, the nodes that have been visited in the previous levels become less important for predicting the potential value of a node in the next level of search. The ES method is hence adopted to guide the search.

To implement the exponential smoothing algorithm in a search agent, we define the following equation and variables:

$$F_{(i,j)} = \alpha RV_{(i-1,k)} + (1 - \alpha)F_{(i-1,k)} \quad (3.12)$$

where $F_{(i,j)}$ is the forecasting value of the j^{th} node in C_i , α is a smoothing constant, $RV_{(i-1,k)}$ is the normalized page value of the k^{th} node in C_{i-1} that points to the j^{th} node in C_i and $F_{(i-1,k)}$ is its forecasting value obtained in search level $i - 1$.

When the search proceeds to the next level, the above equation becomes:

$$F_{(i+1,j)} = \alpha RV_{(i,k)} + (1 - \alpha)F_{(i,k)} . \quad (3.13)$$

The page values The algorithm for implementing the ES algorithm is described in Table 3-7.

In reality, a node is often pointed by many other nodes in a graph (Figure 3-3). Thus, a node can obtain more than one forecasting value. The problem employing the ES algorithm is how to treat all the possible forecasting values of a node in a graph. To solve this problem, we select the maximum one from all these possible values as its forecasting value, which can be illustrated as follows.

Let n_i represent the number of nodes in collection C_i obtained in search level i , $P_{(i+1,j)}$ the j^{th} node in collection C_{i+1} obtained in search level $i+1$. $F_{(i+1,j(i,k))}$ is the forecasting value of node $P_{(i+1,j)}$ obtained from node $P_{(i,k)}$. Assuming node $P_{(i+1,j)}$ is pointed by nodes $P_{(i,1)}, P_{(i,2)}, \dots, P_{(i,n_i)}$ in a graph (Figure 3-5), all the possible forecasting values of node $P_{(i+1,j)}$ are obtained by the following calculations:

$$\begin{aligned} F_{(i+1,j(i,1))} &= \alpha RV_{(i,1)} + (1 - \alpha)F_{(i,1)} \\ F_{(i+1,j(i,2))} &= \alpha RV_{(i,2)} + (1 - \alpha)F_{(i,2)} \\ &\vdots \\ F_{(i+1,j(i,k))} &= \alpha RV_{(i,k)} + (1 - \alpha)F_{(i,k)} \\ &\vdots \\ F_{(i+1,j(i,n_i))} &= \alpha RV_{(i,n_i)} + (1 - \alpha)F_{(i,n_i)} \end{aligned}$$

The forecasting value for node $P_{(i+1,j)}$ is then chosen by the following equation:

$$F_{(i+1,j)} = \text{MAX}(F_{(i+1,j(i,1))}, F_{(i+1,j(i,2))}, \dots, F_{(i+1,j(i,n_i))}) . \quad (3.14)$$

Figure 3-5 graphically illustrates this algorithm.

The above treatment to the forecasting values of a node still leaves room for improvement. Simply selecting the maximum value of a node as the indicator of the search guidance may miss some information. For example, it may be more reasonable to use normalized total forecasting value for each node of next search level. The nodes with higher normalized total forecasting values will represent promising search directions. The details and the further improvement of this algorithm will be studied in our future work.

3.4.2 Size of the Node Set in SWM Using ES algorithm

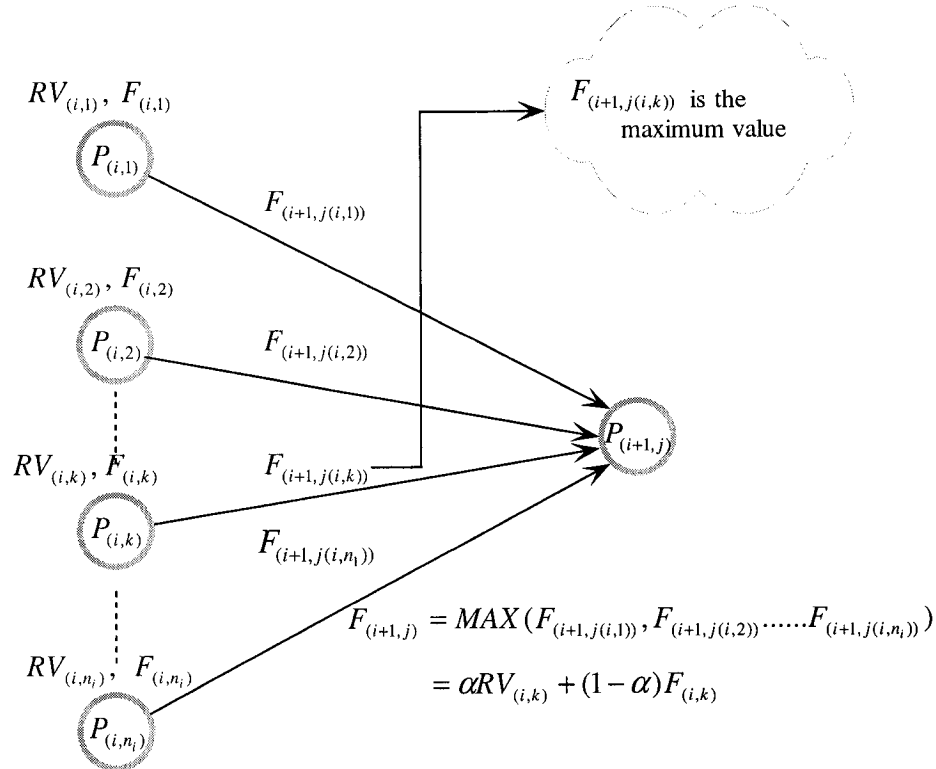


Figure 3-5 Forecasting value of a node

Let U_i represent a sub set of C_i in search level i , U_i is obtained by choosing top m pages in set C_i based on their forecasting values. Because it is infeasible for the search

agent to search all the new nodes obtained from the last search and it is difficult to predefine a threshold RV value to limit the number of pages to be searched in each search level, only top m most promising nodes are taken into consideration in each search level. Therefore, the total node collection S_i for search level i will focus on those nodes with higher forecasting values and is given as follows:

$$S_i = U_i \cup U_{i-1} \cup U_{i-2} \cup \dots \cup U_{i-N_w+1} \quad (3.15)$$

Clearly, the size of the set U_i will be m , and the size of S_i will be $N_w \times m$ if there are no repeated elements in U_i . Figure 3-6 describes the search path and the node set obtained by this approach. In Figure 3-6, the nodes filled with color are with higher forecasting values. In this graph, N_w is 3 and we choose top two nodes in each node collection C_i . Considering node set S_{i+1} , C_{i+1} is the collection of the new nodes obtained in search level

Let n denote the total number of nodes in C_i , m the total number of nodes in C_{i-1} .

$P_{(i,j)}$ the j^{th} node in collection C_i .

For $j=1$ to n

 For $k=1$ to m

 Find the neighbors pointing to $P_{(i,j)}$.

 End for loop.

 /* Forecast the future value of each new node*/

 For $L=1$ to N (the size of the neighbors of $P_{(i,j)}$)

 Calculate the forecasting value of $P_{(i,j)}$ according to

$$F_{(i,j)} = \alpha RV_{(i-1,k)} + (1 - \alpha) F_{(i-1,k)}.$$

 Find the maximum forecasting value of $P_{(i,j)}$.

 End for loop.

End for loop.

Table 3-7 Exponential smoothing algorithm

$i+1$ and the number of the nodes in C_{i+1} is n_{i+1} . After calculating the forecasting values for the new nodes in C_{i+1} using the ES algorithm illustrated in Section 3.4.1, we assume nodes $P_{(i+1,1)}$ and $P_{(i+1,2)}$ obtain higher forecasting values. Therefore, they are the most promising nodes and will be searched in the next level. Hence, these two nodes are added into node set S_{i+1} in this search level for the purpose of calculating the hub and authority values. Obviously, in this case, $U_{i+1} = \{P_{(i+1,1)}, P_{(i+1,2)}\}$. Similarly, we assume $U_i = \{P_{(i,1)}, P_{(i,n_i)}\}$, and $U_{i-1} = \{P_{(i-1,1)}, P_{(i-1,2)}\}$. Then, $S_{i+1} = \{P_{(i-1,1)}, P_{(i-1,2)}, P_{(i,1)}, P_{(i,n_i)}, P_{(i+1,1)}, P_{(i+1,2)}\}$. This way, the node set is limited to a manageable size, thereby reducing the burden in computing the hub and authority values.

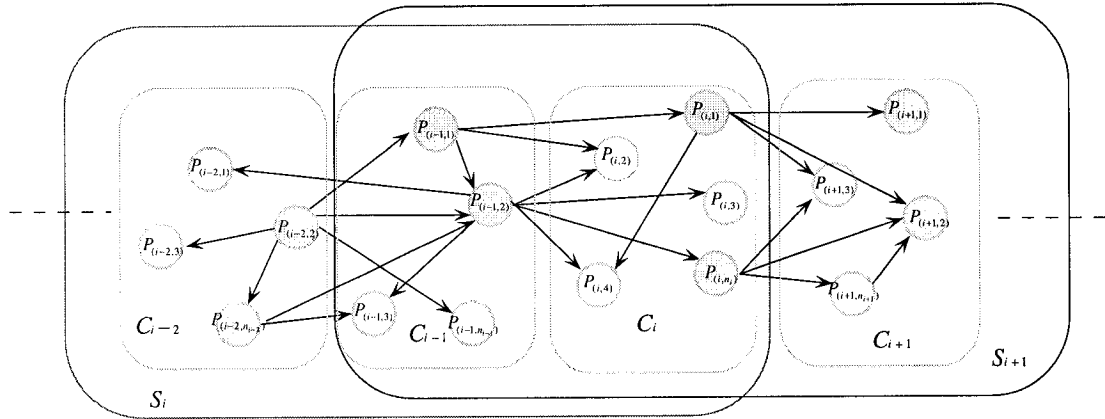


Figure 3-6 Search path and node set controlled by the ES algorithm

3.5 Overall Algorithm of the Search Agent

Figure 3-7 shows the overall algorithm of the search agent implementing the modified TFIDF and ES algorithms in SWM discussed in this chapter. For simplicity, the algorithms of sub functions involved in the implementation of the search agent are omitted.

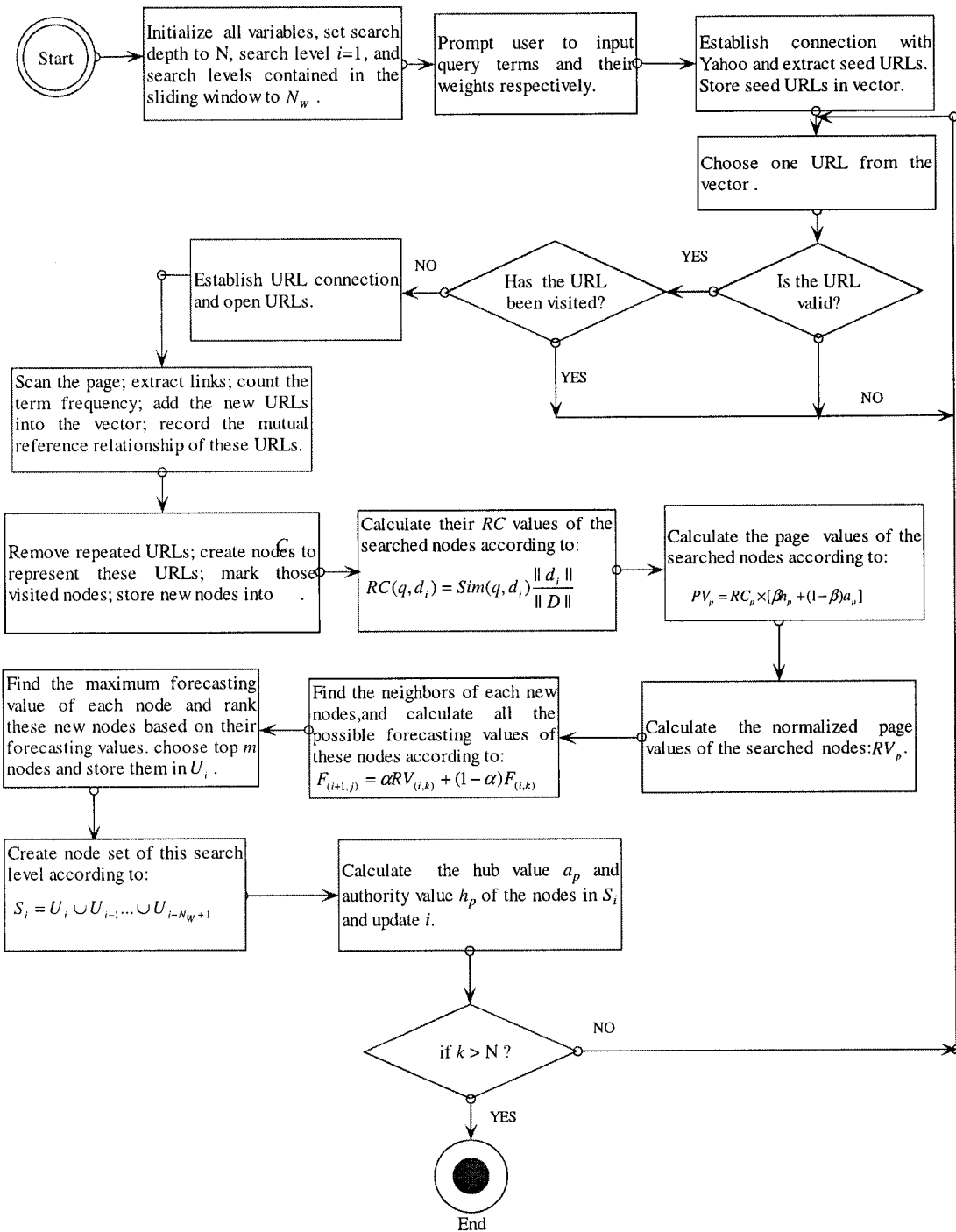


Figure 3-7 Flow chart of the overall algorithm

Chapter 4 Implementation

4.1 Search Depth

The search process can be viewed as a discrete series when the search agent follows the links to look up the information. Hence, we define the search depth and level to mark and control a discrete search series (Mark, 1996). Each level represents a round that the search agent has finished the search of nodes specified in the last level. All these discrete search levels form the depth of a search agent. The user can decide the search depth to control the search process.

4.2 Obtaining the Seed URLs

To start a continuous search, the search agent first needs to collect seed URLs from a popular search engine for a particular topic. In our work, the agent pass query terms through a URL (<http://google.yahoo.com/bin/search?>) to invoke a remote CGI (Common Gateway Interface) residing in the Yahoo server; Yahoo will generate a response to our request, and post the required information back through the connection in the format of byte stream. The agent catches the stream, and extracts the useful URLs contained in it, which is the seed URLs for a continuous search. A sample of the stream returned by Yahoo is listed in Table 4-1.

4.3 Scanning Pages

In this section, we analyze the HTML document structure and describe how the search agent scans the pages and extracts URLs.

A HTML (Hyper Text Mark-up Language) document begins with a DOCTYPE declaration that specifies the version of HTML to which the document conforms. The

HTML element follows and contains the HEAD and BODY. The HEAD contains information about the document, such as its title and keywords, while the BODY contains

```
</big>
&nbsp;- <b>developer</b> information including technical articles, forums, support, bug tracking,
early access program, downloads, and documentation.
<br><font color=006600>http://forte.sun.com/</font>
<br>More sites about:&nbsp;
<font color=blue>
<a
href="http://srd.yahoo.com/srctg/24592/java+developer/2/92/*http://dir.yahoo.com/Business_and_E
conomy/Business_to_Business/Computers/Software/Programming_and_Development/Languages/Jav
a/"><font color=blue>B2B &gt; Programming Software &gt; <b>Java</b></font></a>
</font>
<p>
<li>
<big>
```

Table 4-1 Sample of byte stream returned by Yahoo

the actual content of the document, made up of block-level elements and inline elements. A basic HTML 4.0 document takes on the form that is shown in Table 4-2 (Salton and Buckley, 1996, Soderland,1997, www.htmlhelp.com/reference/html40/structure.html). In the HTML document, a link starts with “<a href>” and ends with “”. Therefore, the contents between the HTML elements “< a href >” and “” suggest a link anchored in this page. Also, we find that the plain text content of this page is embedded between the html tag”>” and “<”, which is a common phenomenon in HTML document presentation. It provides us with an important basis for retrieving the useful plain-text information in this page without retrieving HTML format bytes.

However, there may be a lot of links irrelevant to our query in a web page (they may be advertising links). To avoid collecting such links, we examined the source codes of many

web pages and find another phenomenon: if a link is relevant to our query, it often either contains the query terms in its URL or contains the query terms in the paragraph which are located between this link and the next link, and this paragraph is possibly the summary or excerpts of the page represented by this URL. Although this is not always the case, it greatly reduces the amount of irrelevant URLs and works especially well when the agent extracts useful URLs from Yahoo.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">

  <HTML>
    <HEAD>
      <TITLE>The document title</TITLE>
    </HEAD>
    <BODY>
      <H1>Main heading</H1>
      <P>A paragraph.</P>
      <a href>URI</a>
      <P>Another paragraph.</P>
      <UL>
        <LI>A list item.</LI>
        <LI>Another list item.</LI>
      </UL>
    </BODY>
  </HTML>
```

Table 4-2 HTML document structure

4.4 Primary Search Scenario

In this section, we describe a primary search scenario of the search agent to facilitate the design of the search agent (Table 4-3).

1. Search agent passes our query terms to the remote search engine. Here, we refer to Yahoo.
2. Search agent scans the pages returned by search agent and retrieves the seed URLs.
3. Search agent further opens these seed URLs, scans the content of these pages, and extracts the links and useful information in these pages.
4. Search agent constructs a set of URLs based on the node model, evaluates the values of these URLs, and returns the relative valuable URLs to the user.
5. Search agent forecasts the future values for URLs, and search those URLs with higher forecasting values in priority.
6. Search agent will crawl the web continuously following the links in web pages until it reaches the specified search level.

Table 4-3 Primary search scenario

Based on the above search scenario as well as models and algorithms proposed in Chapter 3, the architecture of the search agent is graphically described in Figure 4-1.

4.5 Class Diagram

To implement such a search agent, we design seven classes by adopting O-O (Object Oriented) programming technique. The source code is written in Java. The reason why we use Java as our programming language is that Java has the enhanced network

programming ability and it provides class library to communicate with the remote server (Vanhelsuwe et al., 1996, Horstman and Cornell, 1998, www.java.sun.com).

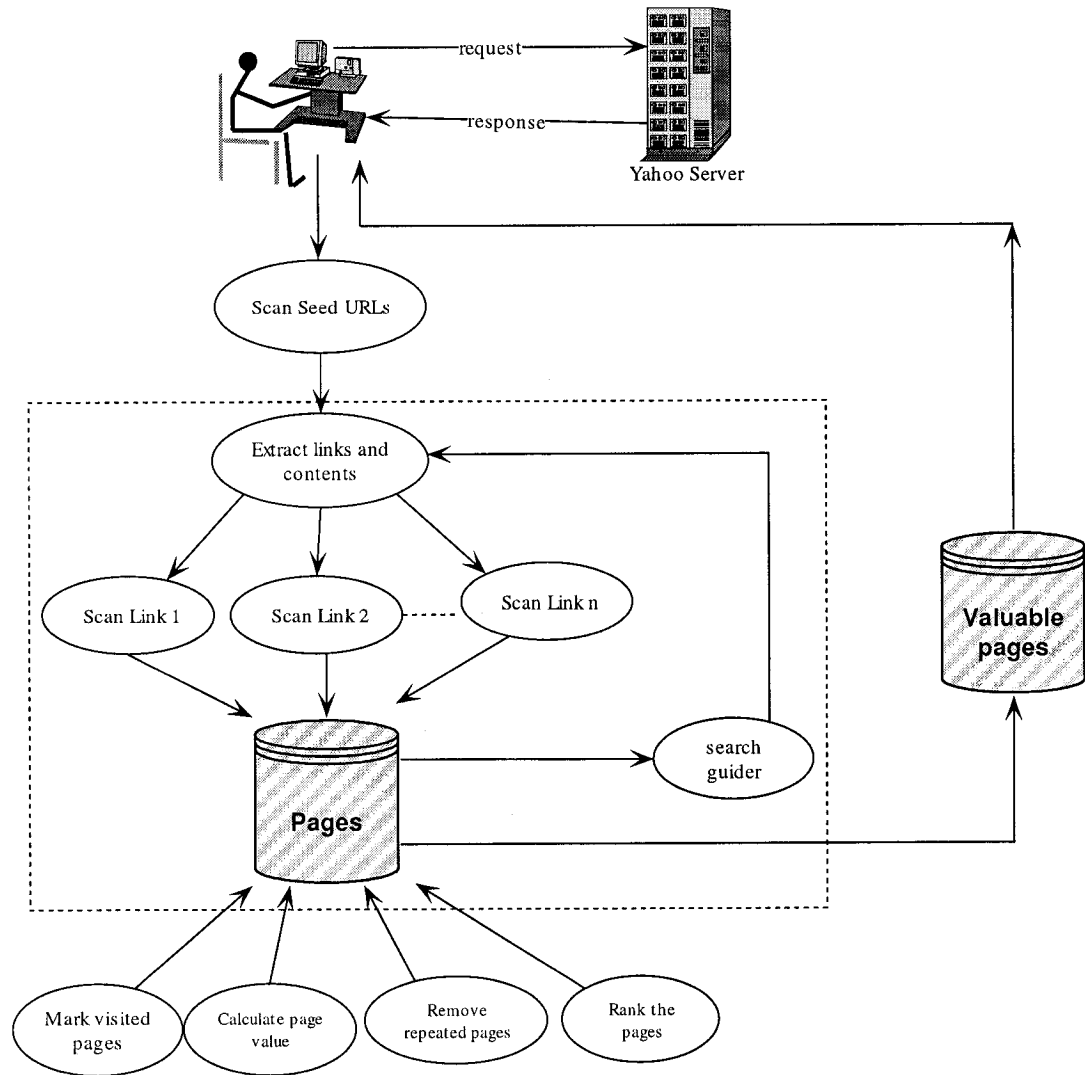


Figure 4-1 Architecture of the search agent

The classes we developed for the search agent are listed below:

- **Scanner class**

This class is mainly responsible for establishing connection with a remote server, catching the byte stream, scanning the contents of the pages, extracting useful links and contents, and removing the repeated URLs it extracted. It also instantiates class `ReadNumber1` to allow user to assign weights for each query term.

- **Vertices2 class**

This class is used to construct a vertex or a node in a directed graph for each link. It will store the hub value, the authority value, the similarity, the *RC* value, the forecasting value as well as the relative value in the node set.

- **Tfidf2 class**

This class is used to implement the modified TFIDF weighting scheme and calculates the similarity value and the *RC* value of each node.

- **Graph2 class**

This class is used to construct a directed graph for the set of vertices (<http://ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/sorting.html> and www.cs.fiu.edu/weiss/cop3530_spr02), calculate the hub value, the authority value, the relative value as well as the forecasting value of each vertex.

- **Baset2 class**

This class is used to build a set of vertices based on our node model.

- **ReadNumber1 class**

This class is used to prompt the user to assign weight for each query term.

- **IndepthSearch11 class**

This class is used to implement a continuous search process.

Figure 4-3 shows the association of these classes. For details on the class diagram, please refer to (http://www.togethersoft.com/services/practical_guides/umlonlinecourse/#class

diagrams).

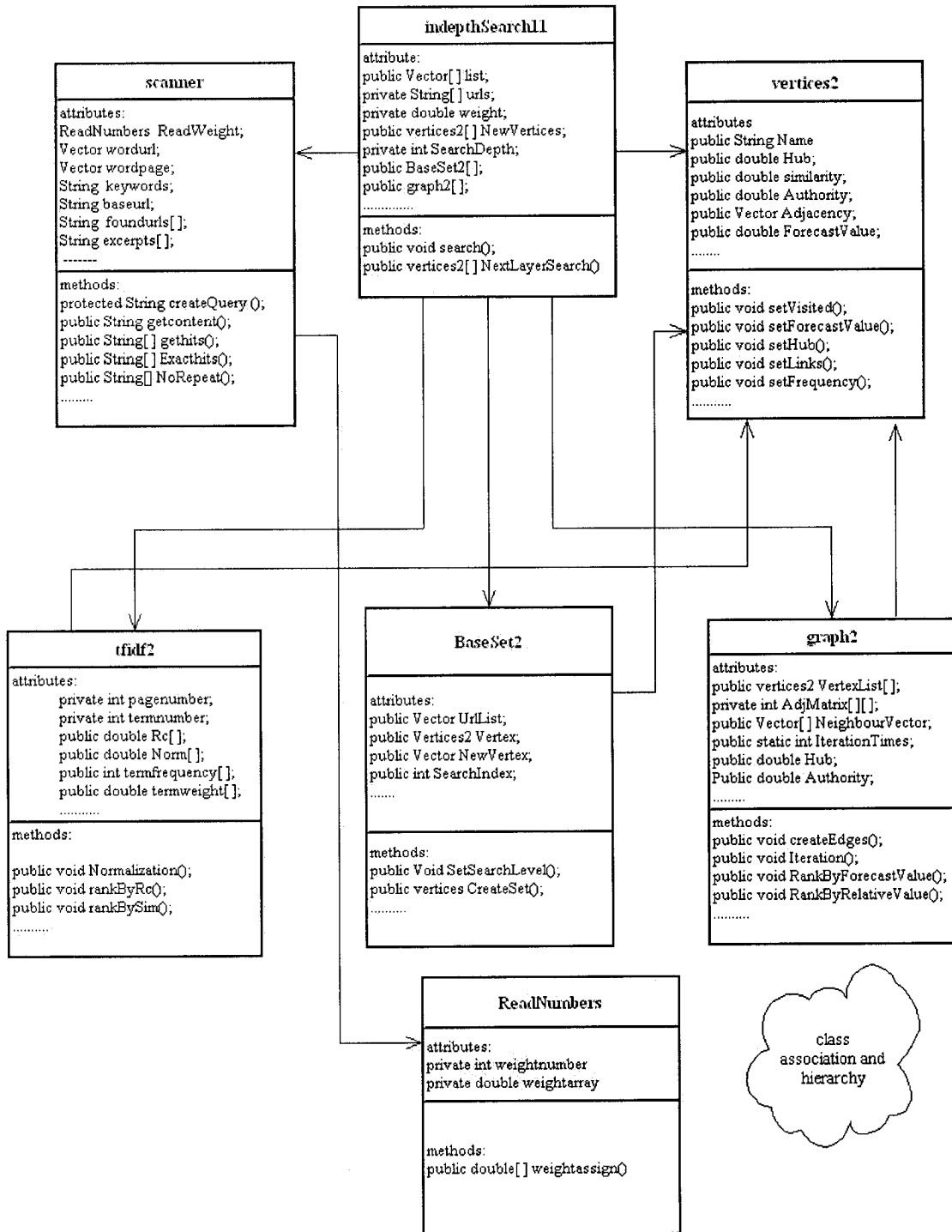


Figure 4-2 Class diagram

Chapter 5 Results and Analysis

In this chapter, we define effectiveness metrics to measure the performance of the proposed models and approaches, and compare the time efficiency and the quality of the obtained information in different cases between different models. We also compare the time efficiency between the ES algorithm and original HITS algorithm proposed by Kleinberg. Because the continuous search process is time-consuming and the iterative algorithm is computationally intensive, we only consider the search scenarios whose search depth is set to eight and the number of query terms is set to two. The effects of content-relevance (*RC*) level are also observed.

5.1 Effects of Content-relevancy Value

In this section, we discuss how the *RC* value defined in Equation 3.1 influences the results retrieved from WWW. In our experiment, the query terms are “java” and “developer” and they are assigned with the same weights. The search URLs that pass our query terms to Yahoo are listed in Table 5-1. The search agent retrieves the first two result pages from Yahoo.

	Search URLs
Page 1	http://google.yahoo.com/bin/search?p=developer+java+&b=0&h=s
Page 2	http://google.yahoo.com/bin/query?p=developer+java&b=20&hc=&hs=&xargs=0
Query terms	Developer, java

Table 5-1 Query terms and search URLs

Figure 5-1 is the screen shot of the first page in Table 5-1. Part of these URLs extracted by the search agent without using TFIDF algorithm is listed in Table 5-2.

The top ten retrieved URLs based on the *RC* value and their corresponding term frequencies are displayed in Figure 5-2. Figure 5-3 shows the top ten retrieved URLs based on the similarity value and their corresponding term frequencies.

The URLs represented by the labels in X-axis of Figure 5-2 and Figure 5-3 are shown in Table 5-3 and Table 5-4 respectively.

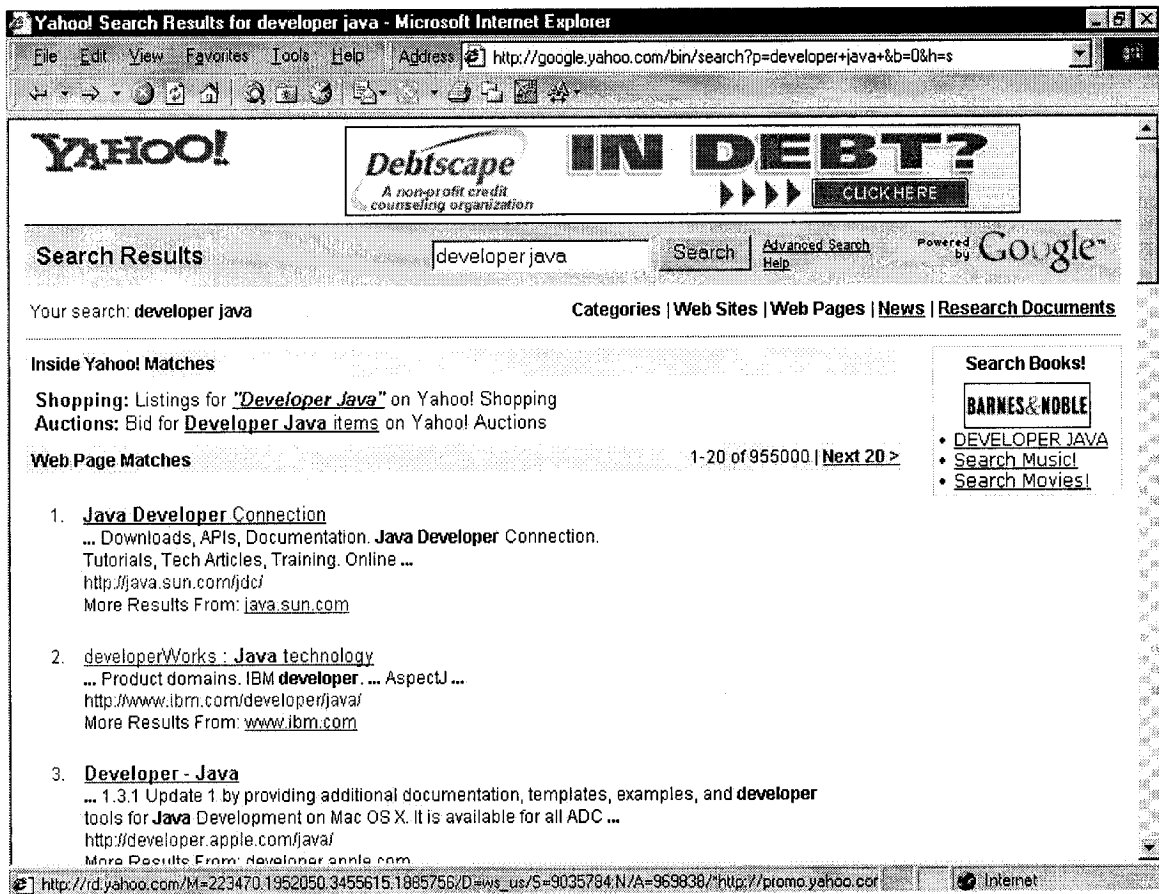


Figure 5-1 Screen shot of the first result page from Yahoo

Comparing Figure 5-2 and Figure 5-3, we see that the URLs in Figure 5-2 have relatively higher term frequencies. This indicates that the modified TFIDF weighting scheme is sensitive to term frequency. As expected, retrieving documents based on the *RC* value can take both the term frequency and the similarity value of a page into consideration and provide better results.

http://java.sun.com/jdc/
http://java.sun.com
http://www.ibm.com/developer/java/
http://www.ibm.com
http://developer.apple.com/java/
http://developer.apple.com
http://builder.com.com/
http://developer.java.sun.com/developer/onlineTraining/
http://developer.java.sun.com
http://wireless.java.sun.com/
http://wireless.java.sun.com
http://webdeveloper.com/java/
http://webdeveloper.com
http://softwaredev.earthweb.com/
http://softwaredev.earthweb.com
http://webdeveloper.earthweb.com/webjs
http://webdeveloper.earthweb.com
http://www.sys-con.com/java/
http://www.sys-con.com
http://www.jars.com/
http://www.jars.com
http://www.entrust.com/developer/java/
http://www.anfyteam.it/
http://forte.sun.com/
http://forte.sun.com
http://www.anfyteam.com/
.....

Table 5-2 Part of the extracted URLs without using the TFIDF algorithm

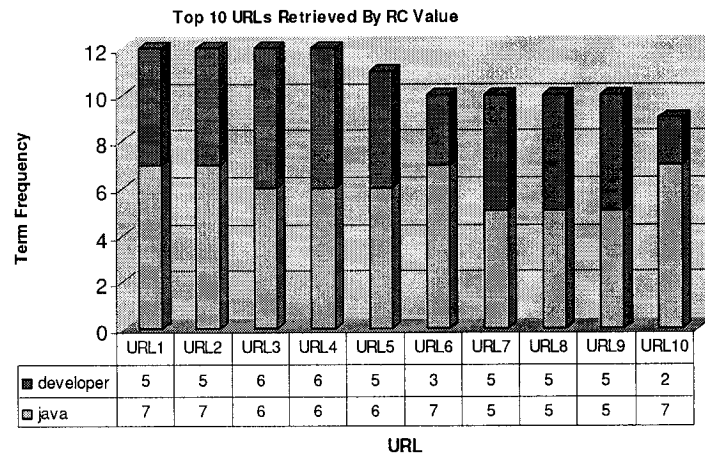


Figure 5-2 Top 10 retrieved URLs based on RC value

Label	URL
URL1	http://industry.java.sun.com/javaneeds/developer/0,2000,,00.html
URL2	http://www-106.ibm.com/developerworks/java/jdk/linux130/?dwzone=java
URL3	http://developer.java.sun.com/developer/onlineTraining
URL4	http://webdeveloper.com/java/
URL5	http://www.javaworld.com/columns/jw-developer-index.shtm
URL6	http://www.techno-link.com/clients/ivo/FreeBuilder/
URL7	http://developer.apple.com/java
URL8	http://www.zdnet.com.au/developer/java
URL9	http://www.sgi.com/developers/devtools/languages/java.html
URL10	http://suned.sun.com/US/certification/java/java_devj2se.html

Table 5-3 The 10 URLs labeled in Figure 5-2

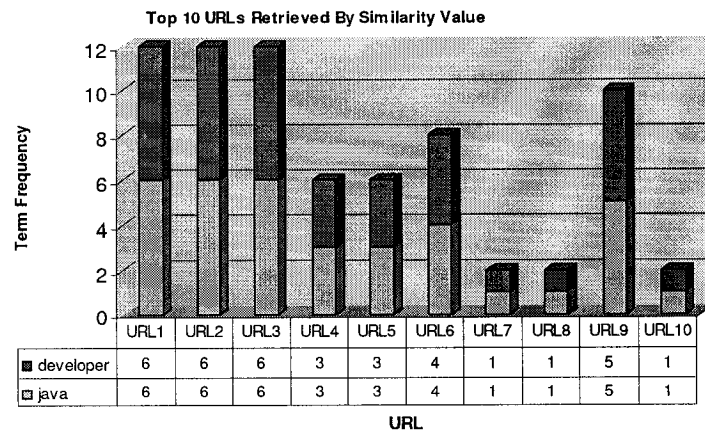


Figure 5-3 Top 10 retrieved URLs based on the similarity

Label	URL
URL1	http://developer.java.sun.com/developer/onlineTraining
URL2	http://developer.java.sun.com
URL3	http://webdeveloper.com/java/
URL4	http://premium.search.yahoo.com/search/premium?p=developer+java+&hc=0&hs=955000
URL5	http://search.auctions.yahoo.com/search/auc?p=developer+java+&clink=dmiyg/developer%2bjava%2b
URL6	http://www.devdaily.com
URL7	http://google.yahoo.com/bin/query?p=developer+java&b=21&hc=&hs=&xargs=0
URL8	http://www.ibm.com
URL9	http://developer.apple.com/java/
URL10	http://www.ubid.com/cat/get_cat_page.asp?CatID=2&s=uwb65064

Table 5-4 The 10 URLs labeled in Figure 5-3

5.2 Search Results and Comparisons

In this section, we demonstrate the application of the proposed search approach using several examples and compare our approach with the original HITS and the Free Expansion Model (FEM). The comparisons are made based on information quality, computing time as well as search speed and they will be defined in Section 5.2.1. Because the agent may collect hundreds of URLs for a specific query in each search level, for the feasibility of experiments, the search agent only search top six pages for each search level, which are ranked based on either the authority value or the forecasting value. For consistency, all the experiments are conducted using the same computer with an Intel Pentium III 930 MHz processor and 256 M RAM. The computer is connected to the remote server through the Business server of Faculty of Administration of University of Ottawa.

5.2.1 Search Effectiveness Metrics

In this section, we define the following effective metrics to compare the performance of the search agent:

- **Information quality**

As shown in Section 3.3, the RV value is a more comprehensive indicator of information quality since it measures both term frequency and similarity between a query and a document as well as the degree of importance of a document. Therefore, we use the total relative value of pages (nodes) TV as a metric for comparison of information quality. TV is defined as follows:

$$TV = \sum_{i=1}^N RV_i \quad (5.1)$$

where N is the number of the searched pages or nodes in a certain search level, RV_i is the relative value of page i in a certain search level. This variable indicates that the higher the TV , the more valuable the pages will be.

- **Total computing time**

In this thesis, we only compare the controllable times that include the time required to compute the hub and authority value and forecasting value. The time required for opening a URL depends on the internet traffic and the access server speed, which is not under the control of our algorithm and hence will not be included for comparison. The computing time is therefore defined as:

$$CT = T_G + T_F \quad (5.3)$$

where T_G is the time of computing the hub value, the authority value and T_F is the time for calculating forecasting value.

5.2.2 Comparisons between the ES and Original HITS

5.2.2.1 Comparison of Information Quality

In this section, we compare the quality of the obtained information by the ES algorithm and the original HITS algorithm using the Sliding Window Model (SWM) in different cases. As discussed in Chapter 3, the ES algorithm calculates the forecasting value of each node in the current level based on the previous RV value, which is a comprehensive indicator, and the forecasting values. The agent employing the ES algorithm will search the top six pages that are ranked in the order of their forecasting values. The original HITS algorithm simply calculates the authority value of the node in the current level and the agent employing the HITS algorithm searches top six pages that are ranked in the order of their authority values. The quality of the obtained information is measured by the total relative value TV in each search level, as defined in Equation 5.1. The test results are plotted in Figures 5-4 to 5-7. The plot with diamond represents the quality of the

information obtained by the search agent employing the ES algorithm and the curve with quadrangle represents the one obtained by the search agent employing the original HITS algorithm. The top six URLs of the first two levels obtained using the two algorithms are listed in tables following each of the figures. For simplicity, only the URLs in the first two search levels are listed in this section and all the URLs the agent has searched in each level and their corresponding *RV* values are listed in Appendix 2.

A) “Automobile Manufacturer” Case

In this case, the query terms are “automobile” and “manufacturer”. Figure 5-4 shows the numeric results. The searched URLs in level one and level two by these two algorithms are listed in Tables 5-5 and 5-6 respectively.

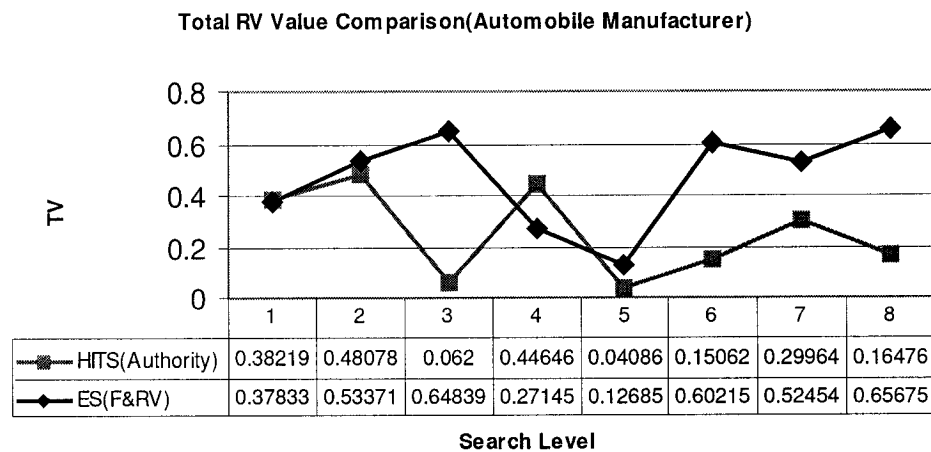


Figure 5-4 Comparison of TV values (Case A)

In Figure 5-4, except in search level 4, the total relative value *TV* obtained by the ES algorithm is higher than that obtained by the original HITS algorithm with an average of 84.9% improvement in information quality.

B) “Java Developer” Case

In this case, the query terms are “java” and “developer”. The *TV* values are summarized in Figure 5-5. Part of the searched URLs is listed in Tables 5-7 and 5-8 respectively.

Level One	
URLs	<i>RV</i>
1. http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1620	0.3783339250091167
2. http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	8.481140106539965E-25
3. http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	8.481140106539965E-25
4. http://www.japanauto.com/	2.1628790645901074E-42
5. http://www.aiam.org/	1.799016066827239E-61
6. http://yp.yahoo.com/py/ypResults.py?stx=manufacturer+automobile	7.1560821094067836E-62
<i>TV</i>: 0.37833	
Level Two	
URLs	<i>RV</i>
1. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search/dir?hc=33&hs=1620&p=manufacturer+automobile	0.3212948911680778
2. http://search.yahoo.com/search/news?hc=33&hs=1620&p=manufacturer+automobile	0.21242099602494113
3. http://alerts.yahoo.com/config/edit_notification?.confirm=n&.done=http://search.news.yahoo.com/search/news?p=manufacturer+automobile&.keywords=manufacturer+automobile&.t=n	4.352576991152729E-19
4. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021025/557846&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dmanufacturer+automobile	4.352576991152729E-20
5. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021024/540099&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dmanufacturer+automobile	4.352576991152729E-20
6. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021010/301442&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dmanufacturer+automobile	4.352576991152729E-20
<i>TV</i>: 0.53371	

Table 5-5 Part of the searched URLs guided by the ES algorithm (Case A)

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1618	0.3821975853094041
2. http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	1.814657518941293E-25
3. http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	1.814657518941293E-25
4. http://dir.yahoo.com/Regional/Countries/Canada/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	5.408295356870514E-34
5. http://www.japanauto.com/	4.639880722685217E-43
6. http://www.iam.org/	3.889058125693456E-62
TV: 0.38219	
Level Two	
URLs	RV
1. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search/dir?hc=33&hs=1618&p=manufacturer+automobile	0.4734973812726609
2. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search?hc=33&hs=1618&p=manufacturer+automobile	0.0070341941815866595
3. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021014/374130&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dmanufacturer+automobile	1.0428568136939265E-9
4. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021010/300101&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dmanufacturer+automobile	1.0428568136939265E-9
5. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021025/557846&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dmanufacturer+automobile	5.214284068469633E-10
6. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021024/540099&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dmanufacturer+automobile	5.214284068469633E-10
TV: 0.48053	

Table 5-6 Part of the searched URLs guided by the HITS algorithm (Case A)

It is clear in Figure 5-5 that the *TV* values obtained by employing ES algorithm are obviously higher than that obtained by the original HITS algorithm. The average improvement of the eight levels is about 170.1%.

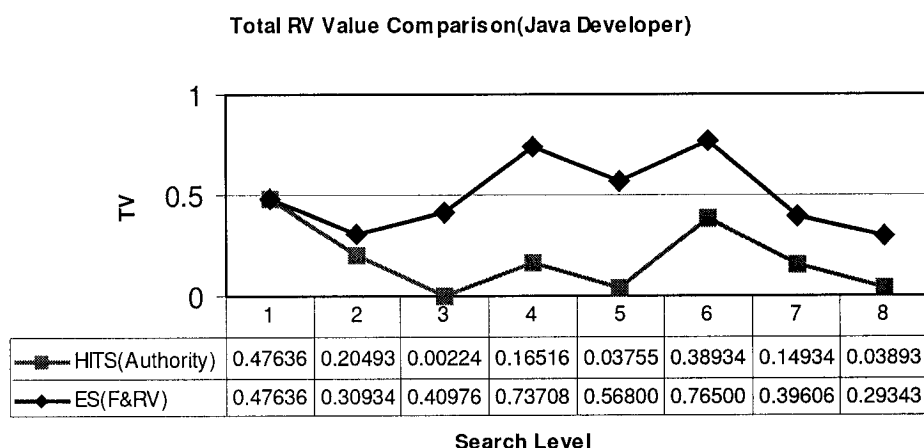


Figure 5-5 Comparison of TV values (Case B)

C) “Artificial Intelligence” Case

In this case, the query terms are “artificial” and “intelligence”. Figure 5-6 shows the *TV* values. The searched URLs in level one and level two by the ES algorithm are listed in Table 5-9. Table 5-10 lists the searched URLs in the first two levels by the original HITS algorithm.

The results in Figure 5-6 show that the ES algorithm outperforms the original HITS algorithm in almost all levels of search. The average improvement is 172.7%.

D) “WWW Conference” Case

In this case, the query terms are “www” and “conference”. The *TV* values are presented in Figure 5-7 and the searched URLs in the first two levels by these two algorithms are listed in Tables 5-11 and 5-12 respectively.

Level One	
URLs	RV
1. http://www.ibm.com/developerworks/java/	0.47591166657281836
2. http://java.sun.com/	4.110937656763211E-4
3. http://developer.java.sun.com	3.92508017136517E-5
4. http://www.javalobby.org/	8.456501281045123E-7
5. http://www.jfind.com/javadev.shtml	1.7595431697454394E-28
6. http://www.javaalmanac.com/	2.624464181625253E-63
TV: 0.47636	
Level Two	
URLs	RV
1. http://www-106.ibm.com/developerworks/java/library/wi-lounge6.html	0.11720037925518405
2. http://www-106.ibm.com/developerworks/ibm/nav.html	0.09114009280963992
3. http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.alphaworks.ibm.com/tech/XI&origin=j	0.0662520324104814
4. http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www7b.boulder.ibm.com/wsdd/&origin=j	0.026255306989342562
5. http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader	0.004519471232810146
6. http://www.jugs.org/CampEclipse/	0.003981325822709766
TV: 0.30935	

Table 5-7 Part of the searched URLs guided by the ES algorithm (Case B)

Level One	
URLs	RV
1. http://www.ibm.com/developerworks/java/	0.47591166657281836
2. http://java.sun.com/ 4	4.110937656763211E-
3. http://developer.java.sun.com	3.92508017136517E-5
4. http://www.javalobby.org/ 7	8.456501281045123E-
5. http://www.jfind.com/javadev.shtml 1.7595431697454394E-28	
6. http://www.javaalmanac.com/ 2.624464181625253E-63	
Level Two	
URLs	RV
1. http://www-106.ibm.com/developerworks/java/library/j-jmx2/	0.15506336341510044
2. http://www-105.ibm.com/developerworks/education.nsf/java-onlinecourse-bytitle/962799E0D39E36FA86256C620054AABE?OpenDocument	0.02812263602099271
3. http://www-106.ibm.com/developerworks/wireless/library/wi-lounge6.html	0.015928298879996774
4. http://www-105.ibm.com/developerworks/education.nsf/java-onlinecourse-bytitle/D79A27E7F24383DB86256C6900505A2A?OpenDocument	0.005155035481318436
5. http://www-106.ibm.com/developerworks/ibm/nav.html	6.25169845247119E-4
6. http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader	3.0882305834292495E-5
TV: 0.20493	

Table 5-8 Part of the searched URLs guided by the HITS algorithm (Case B)

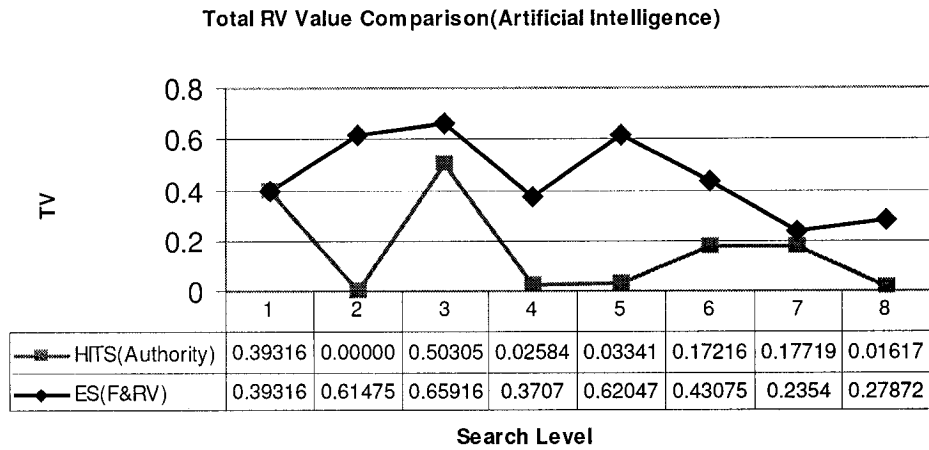


Figure 5-6 Comparison of TV values (Case C)

The numeric results in Figure 5-7 show that the *TV* values obtained by the ES algorithm are higher than that obtained by the original HITS algorithm except in levels 5 and 8, averaging 149.8% improvement in information quality improvement.

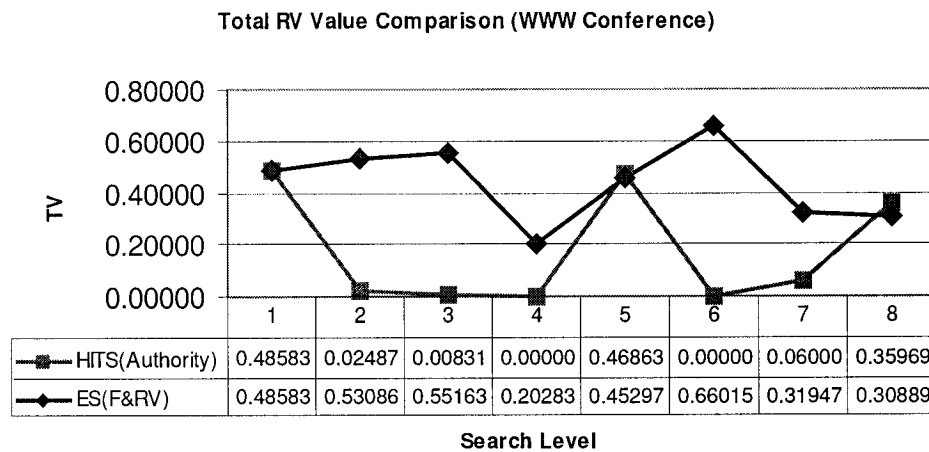


Figure 5-7 Comparison of TV values (Case D)

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=intelligence+artificial+&hc=6&hs=258	0.3931610011405894
2. http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/	6.313229387404063E-18
3. http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/European_Coordinating_Committee_for_Artificial_Intelligence__ECCAL/	2.4163835791280337E-19
4. http://dir.yahoo.com/Entertainment/Movies_and_Film/Titles/Science_Fiction_and_Fantasy/Artificial_Life_Forms/A_I__Artificial_Intelligence/	6.1938976020994054E-27
5. http://dir.yahoo.com/Regional/U_S_States/Massachusetts/Cities/Cambridge/Education/College_and_University/Private/Massachusetts_Institute_of_Technology__MIT_/Departments_and_Programs/Laboratory_for_Computer_Science__LCS_/Artificial_Intelligence_Laboratory/	2.2045883533891348E-35
6. http://www.aaai.org/	8.898778229898588E-52
TV: 0.39316	
Level Two	
URLs	RV
1. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search/dir?hc=6&hs=258&p=intelligence+artificial	0.40366020807461866
2. http://search.yahoo.com/search/news?hc=6&hs=258&p=intelligence+artificial	0.21108695094612345
3. http://alerts.yahoo.com/config/edit_notification?.confirm=n&.done=http://search.news.yahoo.com/search/news?p=intelligence+artificial&.keywords=intelligence+artificial&.t=n	3.6790930111757286E-19
4. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021021/490873&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dintelligence+artificial	3.6790930111757286E-20
5. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021007/242659&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dintelligence+artificial	3.6790930111757286E-20
6. http://premium.news.yahoo.com/news?tmpl=story&u=/nytp/20021021/495620&.bail=http%3a%2f%2fpremium.news.yahoo.com%2frd%3fr%3dintelligence+artificial	3.6790930111757286E-20
TV: 0.61475	

Table 5-9 Part of the searched URLs guided by the ES algorithm (Case C)

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=intelligence+artificial+&hc=6&hs=258	0.3931610011405894
2. http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/	6.313229387404063E-18
3. http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/European_Coordinating_Committee_for_Artificial_Intelligence__ECCAI/	2.4163835791280337E-19
4. http://dir.yahoo.com/Entertainment/Movies_and_Film/Titles/Science_Fiction_and_Fantasy/Artificial_Life_Forms/A_I__Artificial_Intelligence/	6.1938976020994054E-27
5. http://dir.yahoo.com/Regional/U_S__States/Massachusetts/Cities/Cambridge/Education/College_and_University/Private/Massachusetts_Institute_of_Technology__MIT_/Departments_and_Programs/Laboratory_for_Computer_Science__LCS_/Artificial_Intelligence_Laboratory/	2.2045883533891348E-35
6. http://www.aaai.org/	8.898778229898588E-52
TV:0.39316	
Level Two	
URLs	RV
1. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search?hc=6&hs=258&p=intelligence+artificial	0.487004213423207
2. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=42779322/*http://www.rottentomatoes.com/movie-10000163/about.php	2.0810189324318102E-11
3. http://rd.yahoo.com/dir/abl/featuring/?http://movies.yahoo.com/shop?d=hv&cf=info&id=1804383571	2.728592073710427E-12
4. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=27550374/*http://www.dreamworksfansite.com/ai/	2.04644405528282E-12
5. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=58521013/*http://www.filmtracks.com/titles/ai.html	6.821480184276067E-13
6. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=39249876/*http://aimovie.warnerbros.com/	6.821480184276067E-13
TV: 0.48700	

Table 5-10 Part of the searched URLs guided by the HITS algorithm (Case C)

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=conference+www+&c=news	0.4760859446388806
2. http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=27	0.004870644203818917
3. http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=27	0.004870644203818917
4. http://www.iisg.nl/~w3vl/	1.384313307898563E-10
5. http://www.www2002.org/	1.2192502043147239E-26
6. http://www.ftpconferences.com/webdesignworld/	4.4794862095506085E-52
TV: 0.48583	
Level Two	
URLs	RV
1. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search?p=conference+www	0.4875477689930884
2. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search/dir?p=conference+www	0.022191314447732695
3. http://search.yahoo.com/search/news?p=conference+www	0.021122438706080708
4. http://story.news.yahoo.com/news?tmpl=story&u=/cp/20021112/en_bpiep/military_reporters_group_to_hold_1st_conference	1.3931788416257872E-55
5. http://alerts.yahoo.com/config/edit_notification?.confirm=n&.done=http://search.news.yahoo.com/search/news?p=conference+www&.keywords=conference+www&.t=n	4.419657843270001E-56
6. http://rd.yahoo.com/M=239743.2635136.4048612.2395909/D=news/S=8884033:VREC/A=1324285/R=0/*http://service.bfast.com/bfast/click?bfmid=29150849&siteid=39650429&bfpag=account	1.7636989129508945E-56
TV: 0.53086	

Table 5-11 Part of the searched URLs guided by the ES algorithm(Case D)

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=conference+www+&c=news	0.4760859446388806
2. http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=27	0.004870644203818917
3. http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=27	0.004870644203818917
4. http://www.iisg.nl/~w3vl/	1.384313307898563E-10
5. http://www.www2002.org/	1.2192502043147239E-26
6. http://www.ftpconferences.com/webdesignworld/	4.4794862095506085E-52
TV: 0.48583	
Level Two	
URLs	RV
1. http://story.news.yahoo.com/news?tmpl=story&u=/ap/20021107/ap_on_bi_ge/trade_meeting_3	0.004451841783670296
2. http://story.news.yahoo.com/news?tmpl=story&u=/ap/20021108/ap_on_re_eu/archbishop_of_canterbury_1	0.0033772592841636736
3. http://story.news.yahoo.com/news?tmpl=story&u=/ap/20021107/ap_wo_en_po/us_trade_meeting_4	0.0030702357128760665
4. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search?hc=1&hs=26&p=conferences+www	0.0017514710149643144
5. http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search/dir?hc=1&hs=26&p=conferences+www	0.001357334575125027
6. http://story.news.yahoo.com/news?tmpl=story&u=/bv/20021106/co_bv/conferences_make_contacts_for_minority_firms	0.01085922758183334
TV: 0.02487	

Table 5-12 Part of the searched URL guided by the HITS algorithm(Case D)

5.2.2.2 Comparison of Computing Time

In this section, we compare the computing time of the search agent employing the ES algorithm to that of the search agent employing the original HITS algorithm using the SWM in different cases. The computing time is defined in Equation 5.3. The first search agent searches top six pages that are ranked based on their forecasting values and the second search agent searches top six pages that are ranked based on their authority values. For consistency, the experiments are conducted for the same cases discussed in Section 5.2.2.1.

A) “Automobile Manufacturer” Case

The computing times in different search levels in this case are summarized in Figure 5-8.

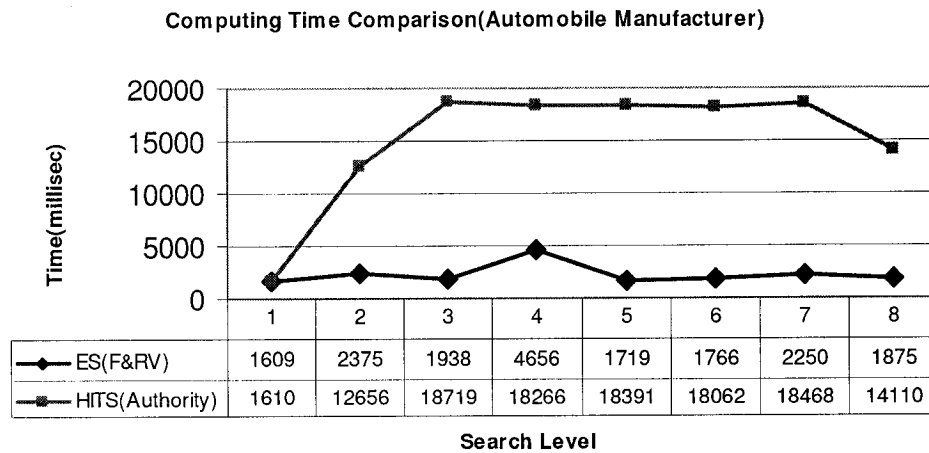


Figure 5-8 Comparison of computing times (Case A)

It is clearly shown in Figure 5-8 that the computing times of the search agent employing the ES algorithm is less than that of the search agent employing the original HITS algorithm. The average computing times obtained by the ES algorithm is only 15.1% of that obtained by the original HITS algorithm.

B) “Java Developer” Case

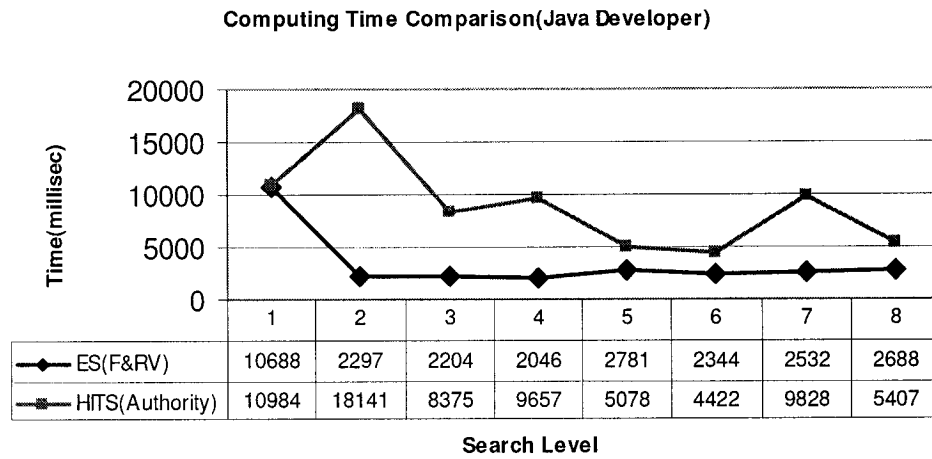


Figure 5-9 Comparison of computing times (Case B)

As expected, the results in Figure 5-9 show that the computing times of the search agent employing the ES algorithm are greatly reduced and are on average 38.4% of that required by the original HITS algorithm.

C) “WWW Conference” Case

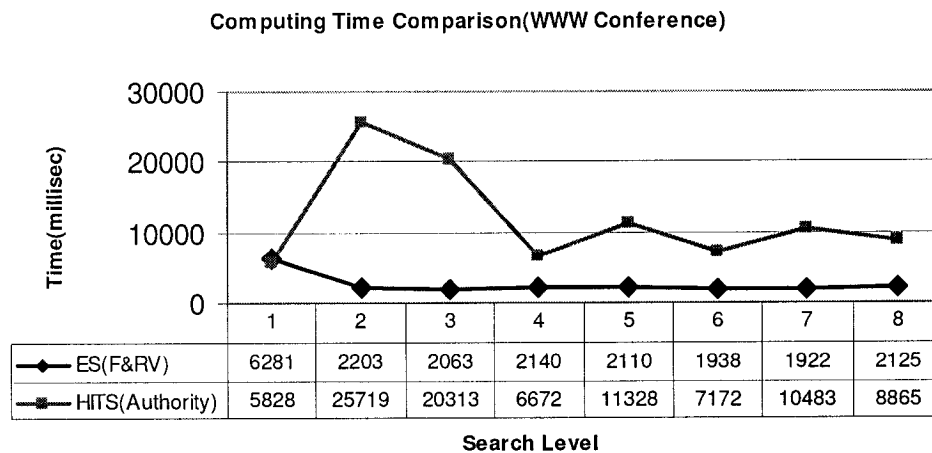


Figure 5-10 Comparison of computing times (Case C)

In Figure 5-10, the computing times of the search agent employing the ES algorithm are obviously less than that obtained using the original HITS algorithm, averaging 21.6 % of that needed by the latter one.

D) “Artificial Intelligence” Case

Figure 5-11 shows the computing times in each search level in this case. The results in Figure 5-11 also show that the search agent employing the ES algorithm has gained much advantage in saving computing times compared to the search agent simply employing the original HITS algorithm. The average computing times obtained by using the ES algorithm are 22.2% of that of the original HITS algorithm.

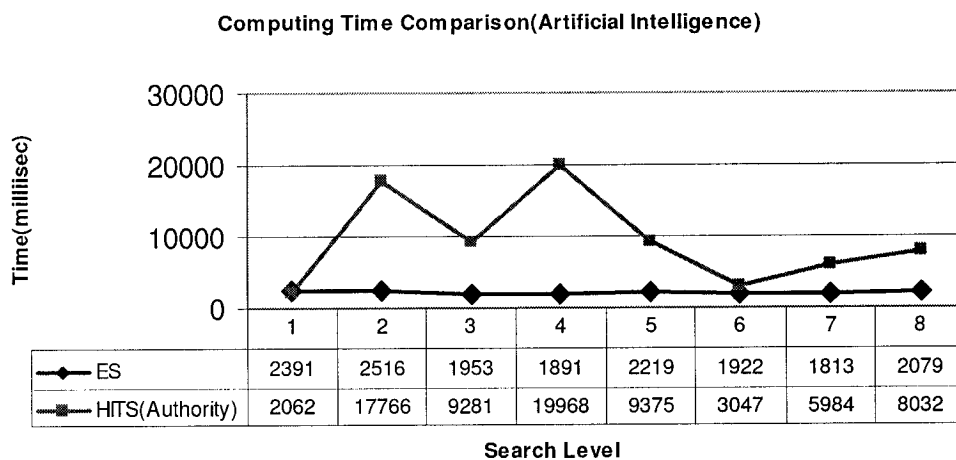


Figure 5-11 Comparison of computing times (Case D)

5.2.3 Computing Time Comparison between FEM and SWM

In this section, we compare the computing times between two different models, FEM and SWM, which are discussed in Chapter 3. The query terms in this experiment are “java”

and “developer”. The search agent that implements SWM or FEM only searches the top six pages in each search level.

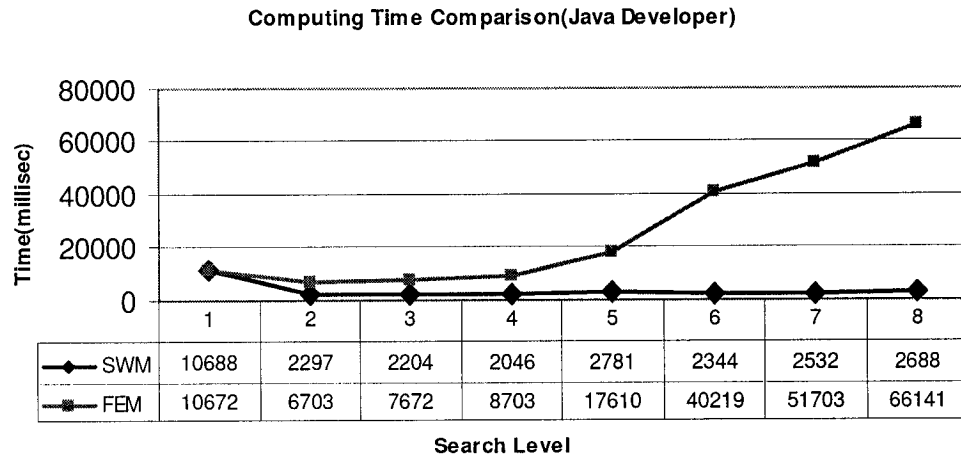


Figure 5-12 Comparison of computing times (SWM vs FEM)

As expected, Figure 5-12 shows that the computing times with the FEM increase exponentially as the search continues, while the computing times with the SWM stabilize at a much lower level. This is because the size of node set in each search level in FEM is continuously increasing as search proceeds while the size of the node set in each search level in SWM is kept almost unchanged. The results illustrate that the proposed approach is much more effective in controlling the computing time compared to FEM.

5.2.4 Effects of Different Smoothing Constants

Different smoothing constants assign different weights to the historic data, and will likely influence the forecasting values. Therefore, it may affect the search path guided by the forecasting value. In this section, we observe how different smoothing constants influence the search process of the search agent. The query terms in this experiment are “java” and “developer”, and the smoothing constants to be used are 0.1, 0.3, 0.5, 0.8 and 0.9 respectively. Only top six pages of each search level are searched based on their forecasting values.

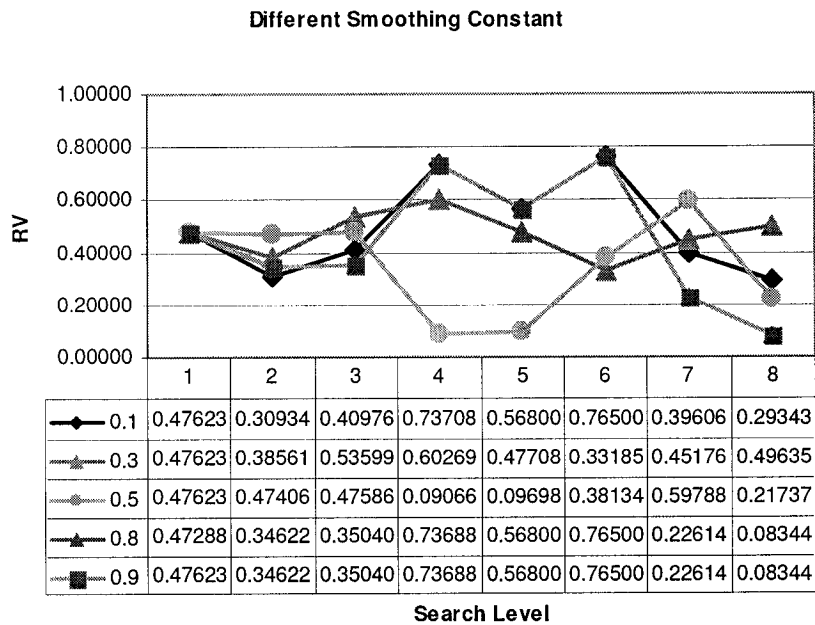


Figure 5-13 Effects of different smoothing constants

In Figure 5-13, both smoothing constants 0.9 and 0.8 provide the same search path, which is indicated by their same *RV* values in each search level. This is because, though the different smoothing constants could affect the forecasting value of a node, they may not necessarily change the order of the nodes ranked by their forecasting values. For other smoothing constants in Figure 5-13, the search agent obtains different *RV* values. Under this situation, the smoothing constants not only affect the forecasting values of the nodes, but also affect their orders ranked by their forecasting values, thus leading to different search paths.

5.2.5 Results Using Modified Page Value Metrics

As can be seen in Tables 5-5 through 5-12, some *RV* values are extremely small. This is caused by the inherent nature of the HITS algorithm. With the HITS algorithm, if a graph is loosely connected, some *RV* values will become very small, and a slight change of the

link relationship can greatly change the hub or authority value of a node. This occurs frequently in Web search and is illustrated as follows:

Considering two simple graphs with six nodes, the hub and authority values of each node computed by the HITS algorithm are listed in Table 5-13.

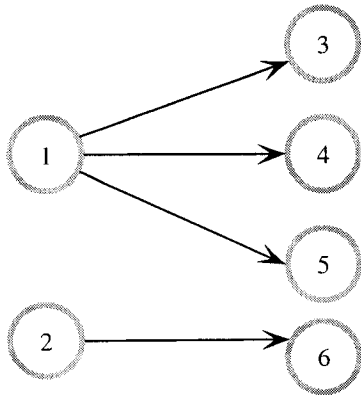


Figure 5-14 A loosely connected graph

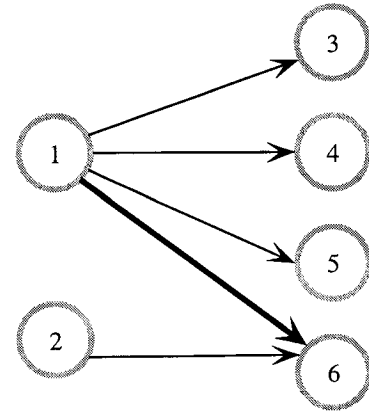


Figure 5-15 A densely connected graph

Nodes \ Graph	Node Values in Figure 5-14		Node Values in Figure 5-15	
	Hub	Authority	Hub	Authority
1	1.0	0.0	0.9571	0.0
2	8.22E-20	0.0	0.2898	0.0
3	0.0	0.57735	0.0	0.4614
4	0.0	0.57735	0.0	0.4614
5	0.0	0.57735	0.0	0.4614
6	0.0	4.75E-20	0.0	0.6011

Table 5-13 Hub and authority values of the nodes in the two graphs

Suppose each node has obtained a RC_p value, according to Equation 3.9, the PV_p and RV_p values of the above two graphs are summarized in Table 5-14, assuming β is 0.1.

Graph Nodes	Node Values in Figure 5-14			Node Values in Figure 5-15		
	RC_p	PV_p	RV_p	RC_p	PV_p	RV_p
1	0.6	0.06	0.03284	0.6	0.05742	0.03160
2	0.8	6.576E-20	3.6013E-20	0.8	0.02318	0.01276
3	1.4	0.72746	0.39824	1.4	0.58136	0.31994
4	1.2	0.62353	0.34134	1.2	0.49831	0.27424
5	0.8	0.41569	0.22823	0.8	0.33221	0.18283
6	0.6	2.565E-20	1.4042E-20	0.6	0.32459	0.17863

Table 5-14 Page values in the two graphs in Figures 5-14 and 5-15

Observing Table 5-13 and Table 5-14, we can see that the hub value of Node 2 and the authority value of Node 6 in Figure 5-14 are virtually zero. As a result, their PV_p and RV_p values are also extremely low compared to the values of the other nodes in this graph. By adding merely a single link between nodes 1 and 6, the hub value of node 2 increases to 0.2898 and the authority value of node 6 raises to 0.6011. This indicates that the HITS based approach is too sensitive to graph structural changes. The RP values calculated in such a way may not be in line with human judgment. For this reason, Equation 3.10 is modified as follows:

$$RP_p = \frac{1}{|\log(\frac{\beta h_p + (1-\beta)a_p}{2})|} \quad (5.3)$$

The new values of the nodes based on the above modification are listed in Table 5-15.

Graph Nodes	Node Values in Figure 5-14				Node Values in Figure 5-15			
	RC_p	RP_p	PV_p	RV_p	RC_p	RP_p	PV_p	RV_p
1	0.6	0.7686	0.46116	0.07275	0.6	0.7621	0.45726	0.06835
2	0.8	0.0491	0.03928	0.00626	0.8	0.5438	0.43504	0.06503
3	1.4	1.7083	2.39162	0.37798	1.4	1.4648	2.05072	0.30656
4	1.2	1.7083	2.04996	0.32338	1.2	1.4648	1.75776	0.26276
5	0.8	1.7083	1.36664	0.21559	0.8	1.4648	0.93184	0.13929
6	0.6	0.0508	0.03048	0.00481	0.6	1.7610	1.0566	0.15794

Table 5-15 Page values after modification

With the modification, the RV_p values in Table 5-15 are more close to human's judgement compared to the page values in Table 5-14. Part of the search results evaluated by the modified metric are listed in Tables 5-16 to 5-23 for the four cases discussed in Section 5.2.2. The rest of the results are listed in Appendix 3.

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1620	0.4223452874385104
2. http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	0.00965370328620864
3. http://dir.yahoo.com/Regional/Countries/Canada/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	0.00826283875052485
4. http://www.japanauto.com/	0.00588277876957169
5. http://www.aiam.org/	0.00468378645210461
6. http://yp.yahoo.com/py/ypResults.py?stx=manufacturer+automobile	0.00239185455486392
TV: 0.4532	
Level Two	
URLs	RV
1. http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile	0.49195917773042336
2. http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile	0.08212736856281715
3. http://yp.search.yahoo.com/search/ypredirect?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile	0.03938170883459406
4. http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D	0.0038421179350692104
5. http://story.news.yahoo.com/news?tmpl=story&u=/nm/20030415/pl_nm/environment_diesel_dc_2	0.001121058967534602
6. http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645	0.000821693780163709
TV: 0.6192	

Table 5-16 New search results guided by the ES algorithm (Case A)

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1620	0.4223452874385104
2. http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	0.009653703286208647
3. http://dir.yahoo.com/Regional/Countries/Canada/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/	0.00826283875052485
4. http://www.japanauto.com/	0.00588277876957169
5. http://www.aiam.org/	0.00468378645210461
6. http://yp.yahoo.com/py/ypResults.py?stx=manufacturer+automobile	0.00239185455486392
TV: 0.4532	
Level Two	
URLs	RV
1. http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile	0.21442 164009203613
2. http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile	0.19381539445954882
3. http://yp.search.yahoo.com/search/ypredirect?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile	0.08975808754671673
4. http://search.news.yahoo.com/search/news/?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile+&c=news_photos	0.039810864163511435
5. http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645	0.008756886589923582
6. http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645	0.006567664942442686
TV: 0.5531	

Table 5-17 New search results guided by the HITS algorithm (Case A)

Level One	
URLs	<i>RV</i>
1. http://www.ibm.com/developerworks/java/	0.4128629605998993
2. http://java.sun.com/	0.03965328753334086
3. http://developer.java.sun.com	0.02427424567471161
4. http://www.javalobby.org/	0.01184420477709548
5. http://www.jfind.com/javadev.shtml	0.00771643412249827
6. http://www.javaalmanac.com/	0.00393804118843172
<i>TV</i>: 0.4999	
Level Two	
URLs	<i>RV</i>
1. http://www-106.ibm.com/developerworks/theme/	0.21389528634262828
2. http://www-106.ibm.com/developerworks/java/library/l-secjav.html	0.20174847712950947
3. http://www-106.ibm.com/developerworks/ibm/nav.html	0.07649925899708516
4. http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www7b.boulder.ibm.com/wsdd/&origin=j	0.0226409369130074
5. http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader	0.01364113199828978
6. http://www.symbian.com/news/2002/expo03-feat.html	0.00287951764014932
<i>TV</i>: 0.5313	

Table 5-18 New search results guided by the ES algorithm (Case B)

Level One	
URLs	RV
1. http://www.ibm.com/developerworks/java/	0.4128629605998993
2. http://java.sun.com/	0.03965328753334086
3. http://developer.java.sun.com	0.02427424567471161
4. http://www.javalobby.org/	0.01184420477709548
5. http://www.jfind.com/javadev.shtml	0.00771643412249827
6. http://www.javaalmanac.com/	0.00393804118843172
TV: 0.4999	
Level Two	
URLs	RV
1. http://www-106.ibm.com/developerworks/java/library/j-jsp04153.html	0.12385042367469752
2. http://www-106.ibm.com/developerworks/java/library/j-jstl0415/	0.0816320479604754
3. http://www-106.ibm.com/developerworks/java/library/j-newviews/	0.048788759036143776
4. http://www-106.ibm.com/developerworks/java/library/j-jtp04223.html	0.04160064859367833
5. http://www-106.ibm.com/developerworks/ibm/nav.html	0.039942197195027083
6. http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader	0.0028765258598279033
TV: 0.3387	

Table 5-19 New search results guided by the HITS algorithm (Case B)

Level One	
URLs	RV
1. http://dir.yahoo.com/Entertainment/Movies_and_Film/Titles/Science_Fiction_and_Fantasy/Artificial_Life_Forms/A_I_Artificial_Intelligence/	0.1886505673185579
2. http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/	0.18073010550157712
3. http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/European_Coordinating_Committee_or_Artificial_Intelligence__ECCAL/	0.07655744993755742
4. http://www.ai.mit.edu/	0.002408330522024467
5. http://www1.overture.com/d/sr/?xargs=02u3hs9yoaj00SvTCChBDFXx0qBc45AxSKk%2FLC2ogXaCEwZghg%2BMxq9Zg2q93YtvvL3nCm83geDEa1UU%2Fptsyf4BZGiVRMIjbbYzDF2HxQAZBbqJQ5QUWQOQBfKLGcLeLDHo8w78Z1jNNiAsml1pWTJcFJcVAjUc0DkRmkCtEuDf419j2as76q3WD9MG9vacb3E3rKGJUFFqxDOpquIP9Dh23%2BPZwbk%2B1ICNknmylOc7%2BBaABV	0.00124 25659234176027
6. http://www.aaai.org/	0.0010729828466247205
TV: 0.4506	

Level Two	
URLs	RV
1. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=42779322/*http://www.rottentomatoes.com/movie-10000163/about.php	0.3581355972297218
2. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=27550374/*http://www.dreamworksfansite.com/ai/6540459276429	0.00632
3. http://srd.yahoo.com/S=245960:D1/CS=245960/SS=42656828/*http://www.singinst.org/	0.006093600122767582
4. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=58521013/*http://www.filmtracks.com/titles/ai.html	0.003017693639517619
5. http://srd.yahoo.com/S=245960:D1/CS=245960/SS=245962/*http://www.aaai.org/	0.0018874133144593935
6. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=39249876/*http://aimovie.warnerbros.com/	0.0012
16582316065081	

Table 5-20 New search results guided by the ES algorithm (Case C)

Level One	
URLs	RV
1. http://dir.yahoo.com/Entertainment/Movies_and_Film/Titles/Science_Fiction_and_Fantasy/Artificial_Life_Forms/A_I_Artificial_Intelligence/	0.1886505673185579
2. http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/	0.18073010550157712
3. http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/European_Coordinating_Committee_or_Artificial_Intelligence_ECCAI/	0.07655744993755742
4. http://www.ai.mit.edu/	0.002408330522024467
5. http://www1.overture.com/d/sr/?xargs=02u3hs9yoaj00SvTCChBDFXx0qBc45AxSKk%2FLC2ogXaCEwZghg%2BMxq9gZg2q93YtvvL3nCm83geDEa1UU%2Fptsyf4BZGiVRMljjbYzDF2HxQAZBbqJQ5QUWQOQBfKLGcLeLDHo8w78Zl1jNNiAsml1pWTJcFJcAju0DkRmkCtEuDf419j2as76q3WD9MG9vacb3E3rKGJUFFqxDOpquiP9Dh23%2BPZwbk%2B1ICNknmylOc7%2BBaABV	0.0012425659234176027
6. http://www.aaai.org/	0.0010729828466247205
TV: 0.4506	
Level Two	
URLs	RV
1. http://www.aaai.org/AITopics/html/current.html	0.3727713512640388
2. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=42779322/*http://www.rottentomatoes.com/movie-10000163/about.php	0.0622171040374019
3. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=27550374/*http://www.dreamworksfansite.com/ai/	0.0061134539539702375
4. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=58521013/*http://www.filmtracks.com/titles/ai.html	0.004075635969313492
5. http://srd.yahoo.com/S=245960:D1/CS=245960/SS=245962/*http://www.aaai.org/	0.0037054488626334244
6. http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=39249876/*http://aimovie.warnerbros.com/	0.002037817984656746
TV: 0.4509	

Table 5-21 New search results guided by the HITS algorithm (Case C)

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=conference+www+&c=news	0.30751 61142110977
2. http://www.sys-con.com/webwest/	0.11375 166226648216
3. http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=31	0.089570527213745507
4. http://www.iisg.nl/~w3vl/	0.01277 8443675237054
5. http://www.www2002.org/	0.00996 7715504353978
6. http://www oulu.fi/~spaceweb/lib/	0.00704 1720594491317
TV: 0.5406	
Level Two	
URLs	RV
1. http://search.yahoo.com/search/dir?c=news&ei=UTF-8&p=conference+www	0.44631382108011247
2. http://yp.search.yahoo.com/search/ypredirect?c=news&ei=UTF-8&p=conference+www	0.08895520277167124
3. http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dconference%2Bwww%2B%26c%3Dnews	0.010107621002417598
4. http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dconference%2Bwww%2B%26c%3Dnews	0.006765269972817512
5. http://rd.yahoo.com/M=224039.1983422.3465437.2905688/D=news/P=m2n1lcf13000500/S=53720272:HEAD/A=1030394/R=0/*http://news.yahoo.com	0.005436305193304572
6. http://rd.yahoo.com/M=243098.2885920.4233620.2839171/D=news/S=53720272:PB/A=1354070/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=243098.2885920.4233620.2839171/D=news/S=53720272:PB/A=1354070/R=5/1051061649+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-rl/http://www.compaq.com/bridge/poweredy/smb/index-yahoo.html	0.0010872610386609145
TV: 0.5586	

Table 5-22 New search results guided by the ES algorithm(Case D)

Level One	
URLs	RV
1. http://search.news.yahoo.com/search/news?p=conference+www+&c=news	0.30751 61142110977
2. http://www.sys-con.com/webwest/	0.11375 166226648216
3. http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=31	0.089570527213745507
4. http://www.iisg.nl/~w3vl/	0.01277 8443675237054
5. http://www.www2002.org/	0.00996 7715504353978
6. http://www oulu.fi/~spaceweb/lib/	0.00704 1720594491317
TV: 0.5406	
Level Two	
URLs	RV
1. http://www2002.org	0.048551716263302314
2. http://www.w3.org	0.010579748743324756
3. http://www.apple.com	0.056204915198912761
4. http://www.research.ibm.com	0.023143200376022903
5. http://www.htdc.org	0.001322468592915594
6. http://www.verity.com	0.001674890334598886
TV: 0.1415	

Table 5-23 New search results guided by the HITS algorithm (Case D)

Chapter 6 Conclusion and Future work

6.1 Conclusion

The proposed models and algorithms discussed in this thesis allow us to gain further insight on information retrieval. In our study, to relieve the user from the drudgery of the search, we designed a search agent which can search web pages on behalf of the user without human supervision; to avoid missing the most relevant pages in a search, we improved the existing TFIDF algorithm and developed a combined link and content based search approach; to maintain a sustainable search and distillation process, we proposed the SWM to control the size of the node set in each search level; to improve the intelligence of the search agent, we employed the ES algorithm in SWM, which was used to guide the search and further control the size of the node set. The experimental results show that: 1) the modified TFIDF algorithm improves the rationality of information retrieval; 2) the ES algorithm in SWM can greatly reduce the computing time and guide the search agent to obtain the information with higher quality.

6.2 Future Work

To make the search agent work more efficiently, future work may focus on the following aspects:

- **Find the proper algorithm and model to improve the speed of content scanning**

The size of the byte stream of a web page tends to be very large. Therefore, scanning a page often involves heavy computation. How to scan a page more efficiently is a challenging task in our future work.

- **Further improve the ES algorithm**

In this thesis, we choose the maximum forecasting value of a node in a graph as its forecasting value. Future work may use the normalized total forecasting value of a node in a graph to guide the search.

- **Improve the validity of the retrieved URLs.**

Some of the retrieved URLs might lack domain information since a link may indicate a relative address in a W3C HTML document structure. Future work should make the search agent perceive an incomplete URL, and recognize the domain for the URL.

- **Build personalized or topic oriented databases for the search results**

To avoid duplicated search effort, a personalized or topic oriented database should be constructed. The previously searched results and search history should be stored in the database. A mechanism should be designed to update the database with minimal search effort.

References

Advanced search, on-line, www.google.com.

Advanced search, on-line, www.yahoo.com.

Advanced search, on-line, www.altavista.com.

Averaging and Exponential Smoothing Models, on-line,
<http://www.duke.edu/~rnau/411avg.htm>.

Balabanovic M., and Shoham Y., 1995, Learning Information Retrieval Agents: Experiments with Automated Web Browsing. *Proceeding of 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*.

Brin S., and Page L., 1998, The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of the Seventh International World Wide Web Conference*, Brisbane, Australia, Elsevier.

Broder A., Kumar R., Maghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A., and Wiener J., 2000, Graph structure in the web, on-line,
<http://www.almaden.ibm.com/cs/k53/www9.final/>.

Chakrabarti S., 2001, Integrating the Document Object Model with Hyperlinks for Enhanced Topic Distillation and Information Extraction. *Proceedings of the tenth international conference on World Wide Web*, pp. 211-220, New York: ACM .

Chakrabarti S., Dom B. E., Gibson D., Kleinberg J. M., Kumar R., Raghavan P., Rajagopalan S., and Tomkins A., 1999, Mining the Link Structure of the World Wide Web, on-line, www.almaden.ibm.com/cs/k53/ieeecomps.ps.

Chen L., and Sycara K., 1998, WebMate: A Personal Agent for Browsing and Searching. *Proceedings of the second international conference on Autonomous agents*, pp.132-139, New York: ACM Press.

Data Structure and Algorithms, on-line,
<http://ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/sorting.html>.

Data Structure, on-line
www.cs.fiu.edu/~weiss/cop3530_spr02.

Dean J., and Henzinger M.R., 1999, Finding related Pages in the World Wide Web. on-line, <http://www.unizh.ch/home/mazzo/reports/www8conf/2148/pdf/pd1.pdf>.

Deo N., and Gupta P., 2001, World Wide Web: A Graph-Theoretic Perspective, *Computer Science Technical Report CS-TR-01-001*, School of Computer Science, University of Central Florida.

Engineering statistics handbook, on-line,
<http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc43.htm>.

Features in Copernic 2001, on-line, www.copernic.com.

Lawrence S., and Gilles C. L., 1998, Searching the World Wide Web, on-line,
<http://www.neci.nec.com/~lawrence/science98.html>.

Flake W., Lawrence S., and Giles C. L., 2000, Efficient Identification of Web Communities. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 150-160, New York: ACM.

Glover E. J., Lawrence S., Gordon M. D., BirminGhAM W. P., and Giles C. L., 2001, Web Search-Your Way. *Communications of the ACM*, 44(12), pp. 97-102.

Golub G., Van L. C. F., 1989, *Matrix Computations*, Johns Hopkins University Press.

Graph, on-line, <http://www.nist.gov/dads/HTML/graph.html>.

Hersovici M., Jacovi M., Marrek Y. S., Shtalhaim D. P. M., and Ur S., 1998, The shark-search algorithm-an application: tailored web site mapping, *Proceedings of the seventh international conference on World Wide Web 7*, pp. 317-326, Amsterdam: Elsevier Science.

Horstmann C. S., and Cornell G., 1998, *Core JAVA*, Volume 2, *Networking*, pp. 133-176. Palo Alto, California.

Java API, on-line, www.java.sun.com.

Java Programming, on-line,
<http://www.scs.carleton.ca/~courses/1005/Notes/index.html>.

Jones K. S., 2001, course notes, on-line,
<http://www.cs.cornell.edu/courses/cs430/2001fa/syllabus.html>.

Kleinberg J. M., 1999, Authoritative Sources in a Hyperlinked Environment, *Journal of the ACM*, Vol.5 (46), pp. 604-632.

Kleinberg J. M., and Lawrence S., 2001, The Structure of the Web. on-line,
<http://www.sciencemag.org/cgi/content/full/294/5548/1849>.

Lee D, Huei C. L., and Seamons, K. 1997, Document Ranking and the Vector Space Model. *IEEE Software*, Vol.14 (2), pp. 67-75.

Lewis D. D., 1996, Representation and Learning in Information Retrieval. *Ph.D thesis*. Amherst, MA: University of Massachusetts.

Loke S. W., Davion A., and Leon S., 1996, An intelligent Agent for Citation Finding on the World Wide Web. *Proceedings of the 4th Pacific Rim International Conference on Artificial Intelligence (PRICAI'96)*, Cairns, Australia, pp. 580-591.

Mark W., 1996, *Intelligent Java Applications for Internet and Intranet, Infoagent*.

Michtell T., 1999, Machine Learning and Data Mining, *Communications of the ACM*, Vol. 42 (11).

Moukas A., and Maes P., 1998, An evolving multi-agent information filtering and discovery system for the WWW, *Autonomous Agents and Multi-agent Systems*, Vol.1 (1).

Moukas A., 1996, Amaltheae: Information Discovery and Filtering using a Multi-agent Evolving Ecosystem, *Practical Application of Intelligent Agents & Multi-Agent Technology*, London.

Nahm U. Y., and Raymond J. M., 2001, Mining Soft-Matching Rules from Textual Data, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, Seattle, WA.

Nahiam, S., 2001, *Prediction and operations analysis*, 4th ed., McGraw-Hill Irwin, Boston.

Practical UML: A Hands-On Introduction for Developers, on-line
http://www.togethersoft.com/services/practical_guides/umlonlinecourse/#classdiagrams.

Principles of Information Retrieval, course notes on-line,

http://www.sims.berkeley.edu/courses/is240/s02/Lectures/Lecture7_240.ppt.

Salton G., 1989, Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Series in Computer Science. Addison-Wesley Publishing Company, Inc.

Salton G., and Buckley C., 1996, Term Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management* Vol.32 (4), pp. 431-443.

Salton G., 1988, *Automatic Text Processing*. Addison-Wesley Publishing Company.

Structure of a HTML-4.0 document, on-line,

<http://www.htmlhelp.com/reference/html40/structure.html>.

Search features, on-line, www.searchenginewatch.com.

Soderland S., 1997, Learning to Extract Text-based Information from the world wide web, *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 251-254, Newport Beach, California.

Sunny L., 2001, The Overview of Web Search Engines, on-line,

<http://db.uwaterloo.ca/~tozsu/courses/cs748t/surveys/sunny.pdf>.

The global structure of an HTML document, on-line,

<http://www.w3.org/TR/REC-html40/struct/global.html>.

Vanhelsuwe L., Phillips I., and Hsu G. T., 1996, *Mastering Java*, SYBEX, Network Programming, pp. 673-719. Alameda USA.

Vector space model, on-line

<http://isp.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html>.

Yuhono B., and Lee D. L., 1996, Search and Ranking Algorithms for Locating Resources on the World Wide Web. *Proceedings of the IEEE International Conference on Data Engineering* , pp. 164-177, New Orleans, LA.

Zipf Curves and Website Popularity, on-line,
<http://www.useit.com/alertbox/zipf.html>.

Zipf's Law, on-line, <http://linkage.rockefeller.edu/wli/zipf/>.

Appendices

Appendix 1 Source Code

```
*****
*
*
*   The source code of an intelligent search agent
*
*
*
*           Written by: Yongsheng Liu
*
*           Supervised by: Dr. Ming Liang
*
*****
```

To run this program in Dos Command Mode, User must:

1. Compile the main class. Type: "javac TestIndepthSearch10.java" and press "Enter" key.
2. Execute the program. Type: "java TestIndepthSearch10 queryterms". Here queryterms are terms specified by the user. Query terms must be separated by space.
3. Enter weight for each term. The program will prompt user to enter weight for each query term to reflect the user's preference. Enter any real number for each term separated by space.

```
/*this class is used to carry out a continuous search*/
import java.io.*;
import java.net.*;
import java.util.*;
import java.lang.*;

class indepthsearch11
{
    String[] searchwords;
    /* Where we are going to store the processed items */
    public Vector[] list;
    private String[] urls;
    public String Baseurl;
    private vertices2 processing;
```

```

private Scanner pagescan1;
//private Scanner pagescan2;
private vertices2[] URLvertex;
private Vector VertexList;
private tfidf FTFIDF;
private int termnumber;
private Vector[] termcount;
private int[][] termfrequency;
private String[] summary;
private double[] weight;
private Vector N;
private static final int TOPN=6;
private int pagenumber;
private BaseSet2 FSet;
public vertices2[] NewVertices;
private tfidf2 TFIDF2;
private double Sim[];
private double RC[];
private int SearchDepth=7;
/** The MAXIMUM number of links to find */
public static final int MAX_LINKS = 100;
public vertices2[] Vertex;
public tfidf2[] TFIDF;
public Vector[] SearchInfo;
public BaseSet2[] Set;
public graph2[] NewGraph;
private int TopNumber=20;
public Vector[] Top;
public int size;

/* Construtor of this class */

public indepthsearch11 ( String[] keywords) throws IOException
{
    TFIDF=new tfidf2[SearchDepth];
    SearchInfo=new Vector[SearchDepth+1];
    this.list=new Vector[SearchDepth+1];
    for(int i=0;i<SearchDepth+1;i++){
        SearchInfo[i]=new Vector();
        list[i]=new Vector();
    }
    Set=new BaseSet2[SearchDepth];
    NewGraph=new graph2[SearchDepth];
    this.searchwords = keywords;
    this.pagescan1=new Scanner(keywords);
    this.urls=pagescan1.foundurls;
    this.VertexList=new Vector();
    pagenumber=pagescan1.Collection;
    termnumber=pagescan1.keywords.length;
    termcount=pagescan1.wordcount;
    termfrequency=new int[termnumber][pagenumber];
    /* build the termfrequency matrix for all the pages*/
    for(int i1=0;i1<termnumber;i1++){
        int temp=pagescan1.wordcount[i1].size();
        for(int k=0;k<pagenumber;k++){

```

```

        for(int j=0; j<temp;j++){
            System.out.println("the pagenumber is
            "+pagenumber );
            System.out.println("the No repeat url index size is
            "+pagescanl.NoRepeatUrlIndex.size());
            Integer
temp1=(Integer)pagescanl.NoRepeatCountIndex[i1].elementAt(j);
            Integer
temp2=(Integer)pagescanl.NoRepeatUrlIndex.elementAt(k);
            if(temp2.intValue()==temp1.intValue()){
                Integer
temp3=(Integer)pagescanl.wordcount[i1].elementAt(j);
                termfrequency[i1][k]=temp3.intValue();
                break;
            }
            else{
                termfrequency[i1][k]=0;
            }
        }
    }
    System.out.println("wordcount element:
    "+pagescanl.wordcount[i1]);
}

this.summary=pagescanl.excerpts;
weight=pagescanl.queryweight;
N=pagescanl.wordpage;
this.Baseurl=pagescanl.baseurl;
for(int k=0; k<termnumber;k++){
    System.out.println("the queryweight is"+ weight[k]);
    System.out.println("the wordpage is"+ N.elementAt(k));
}
/* Calculate the similarity and RC value for each page */
FTFIDF = new tfidf(pagenumber,termnumber,
termfrequency,weight,N,urls,summary);
FTFIDF.Normalization();
this.urls=FTFIDF.rankByRc();
this.Sim=FTFIDF.sim;
this.RC=FTFIDF.Rc;
FTFIDF.writeout();
this.summary=FTFIDF.HitSection;
}

public static void main( String args[] ) throws IOException
{
    indepthsearch11 indepth = new indepthsearch11(args);

    indepth.search();
}
/* scan a page */
public void search(){
    for ( int i = 0; i < urls.length; i++ ){
        String s = urls[i];

```

```

processing=new vertices2(urls[i],null,termnumber);
processing.setSummary(summary[i]);
Vector[] NoRepeatcountIndex=new Vector[termnumber];
for(int j=0;j<termnumber;j++){
    NoRepeatcountIndex[j]=new Vector();
}
Vector NoRepeatUrlIndex=new Vector();
Vector Temp=new Vector();
String content = pagescan1.getcontent(s);
int checklink=content.indexOf("<a href=");
if(checklink== -1)
continue;

/* Clear the vectors before storing new values */
for(int k=0;k<termnumber;k++){
    pagescan1.wordcount[k].clear();
    pagescan1.wordurl[k].clear();
    pagescan1.NoRepeatCountIndex[k].clear();
    pagescan1.wordurlIndex[k].clear();
    pagescan1.NewWordUrlIndex[k].clear();
}
pagescan1.wordpage.clear();
pagescan1.NoRepeatUrlIndex.clear();
pagescan1.newHits.clear();
Vector pagecollection=new Vector();
String[] linkurls=pagescan1.gethits(content);
linkurls=pagescan1.Exacthits(linkurls);
linkurls=pagescan1.NoRepeat(linkurls);
pagescan1.countpage();
pagenumber=linkurls.length;
termcount=pagescan1.wordcount;
pagecollection=pagescan1.wordpage;
NoRepeatUrlIndex=pagescan1.NoRepeatUrlIndex;
NoRepeatcountIndex=pagescan1.NoRepeatCountIndex;
termfrequency=new int[termnumber][pagenumber];
if(pagenumber!=0){
    for(int il=0;il<termnumber;il++){
        try{

            boolean append=true;
            DataOutputStream Fout = new DataOutputStream(new
                FileOutputStream("outputFrequency.dat", app
                    end));

            String format="\n##### THE FREQUENCY
#####"+ "\n\n";
            Fout.writeUTF(format);
            String temp5="\n*****"+s+"*****";
            String temp3="\n NO REPEAT COUNT INDEX
"+NoRepeatcountIndex[il]+"====="+termcount[il];
            byte[] format5=temp5.getBytes();
            byte[] format3=temp3.getBytes();
            Fout.write(format5);
            Fout.write(format3);

```

```

        String temp4="\n"+"NO REPEAT URL INDEX
"+NoRepeatUrlIndex+"-----\n";
        byte[] format4=temp4.getBytes();
        Fout.write(format4);
        Fout.close();
    }

    catch(IOException excpt){
        System.out.println("error occurs when output to file");
    }
    int temp=termcount[i1].size();
        for(int k=0;k<pagenumber;k++){
            for(int j=0; j<temp;j++){
                System.out.println("the
pagenumber is "+pagenumber );
                System.out.println("the No repeat
url index size is
                "+NoRepeatUrlIndex.size());
                Integer
temp1=(Integer)NoRepeatcountIndex[i1].elementAt(j);
                Integer
temp2=(Integer)NoRepeatUrlIndex.elementAt(k);

                if(temp2.intValue()==temp1.intValue()){
                    Integer
temp3=(Integer)termcount[i1].elementAt(j);

                    termfrequency[i1][k]=temp3.intValue();
                    break;
                }
                else{
                    termfrequency[i1][k]=0;
                }
            }
        }
        System.out.println("wordcount element:
"+pagescanl.wordcount[i1]);
    }

    try{
        boolean append=true;
        DataOutputStream Fout = new
        DataOutputStream(new
        FileOutputStream("MatrixFrequency.dat", append));

        String format="\n##### THE
MATRIX FREQUENCY STATISTICS #####"+ "\n\n";
        Fout.writeUTF(format);
        for(int k=0;k<termnumber;k++){
            for(int i1=0;i1<pagenumber;i1++){

                String temp6="\n"+"the term
frequency is "+termfrequency[k][i1]+"==";

```

```

        byte[] format6=temp6.getBytes();
        Fout.write(format6);
    }
    }
    Fout.close();
}

catch(IOException excpt){
System.out.println("error occurs when output to file");
}

String[] Nextsummary=new
String[pagescan1.newHits.size()];
pagescan1.newHits.copyInto(Nextsummary);
processing.setN(pagecollection);
processing.setPageNumber(pagenuumber);
processing.setLinkTermFrequency(termfrequency);
processing.setFrequency();
processing.setLinkSummary(Nextsummary);

for(int j=0;j<linkurls.length;j++){
    Temp.addElement(linkurls[j]);
}
processing.setLinks(Temp);

SearchInfo[0].addElement(processing);
VertexList.addElement(processing);
// If the page contained any links
if ( linkurls != null )
{
    for ( int k = 0; k < linkurls.length; k++ )
    {
        if ( linkurls[ k ] != null )
            list[0].addElement(linkurls[ k ] ) ;
    }

    System.out.println( s );
}

}
if ( list[0].size() > 200 )

break;
}

URLvertex=new vertices2[VertexList.size()];

VertexList.copyInto(URLvertex);
try{
    boolean append=true;
    DataOutputStream Fout = new DataOutputStream(new
FileOutputStream("outputURLvertex.dat", append));

```

```

        String aFormat="\n##### THE URL
VERTEX#####"+"the URL vertex number"+URLvertex.length;
        Fout.writeUTF(aFormat);
        for(int i=0;i<URLvertex.length;i++){
            for (int k=0;k<URLvertex[i].pagenumber;k++){
                for(int j=0;j<termnumber;j++){
                    int[] frequency=URLvertex[i].Frequency;
                    String temp="\nthe processed URL frequency
"+frequency[j]+"the vertex name"+URLvertex[i].Name;
                    //URLvertex[i].Frequency[j];
                    byte[] format=temp.getBytes();
                    Fout.write(format);
                }
                for(int j=0;j<termnumber;j++){
                    int[][]
                    linkfrequency=URLvertex[i].LinkTermFrequency;
                    String temp2="\n\nthe link
termfrequency"+linkfrequency[j][k];
                    byte[] format2=temp2.getBytes();
                    Fout.write(format2);
                }
            }
        }
        Fout.close();
    }
    catch(IOException excpt){
        System.out.println("error occurs when output to
file"+excpt);
    }
    try{
        boolean Tappend=true;
        String FileName="TotalComputingTime"+"["+1+"]"+" .txt";
        DataOutputStream FTout = new DataOutputStream(new
        FileOutputStream(FileName,Tappend));
        Date Ctime=new Date();

        String aTFormat="\n#####The Start Time of This
computation #####"+"\n"+Ctime.getDate()+" "+Ctime.getHours()
        +": "+Ctime.getMinutes()+" "+Ctime.getSeconds()+"
        \n";
        FTout.writeUTF(aTFormat);
        FSet=new BaseSet2(URLvertex,termnumber);

        /* create a base set for these urls */
        FSet.SetSearchLevel(1);
        FSet.SetNewSearchUrl(list[0]);
        FSet.CreateLevelVertex();
        FSet.CreateSet();
        NewVertices=FSet.Vertex;
        System.out.println("the number of new
vertices"+NewVertices.length);
        String[] NewVerticesName=FSet.VerticesName;
        int numberOfVertices=NewVertices.length;
        try{
            boolean append=true;

```

```

        DataOutputStream Fout = new DataOutputStream(new
        FileOutputStream("outputBaset1.dat", append));
        String aFormat="\n##### THE URL
VERTEX#####"+
        "\n\nThe Base Url";
        Fout.writeUTF(aFormat);
        for(int i=0;i<numberOfVertices;i++){
            for(int j=0;j<termnumber;j++){
                int[] frequency=NewVertices[i].Frequency;
                String temp="\nthe processed URL frequency
"+frequency[j]+"the vertex
name"+NewVertices[i].Name;//URLvertex[i].Frequency[j];
                byte[] format=temp.getBytes();
                Fout.write(format);
            }
        }
        Fout.close();
    }
    catch(IOException excpt){
        System.out.println("error occurs when output to
file"+excpt);
    }

    TFIDF2=new tfidf2(numberOfVertices,termnumber,weight,NewVertices);
    TFIDF2.SetSearchLevel(1);
    TFIDF2.Normalization();
    TFIDF2.writeout();
    System.out.println("the new vertex length is
"+FSet.NewVertex.size());
    /* Create a graph for these urls */
    graph2 NewGraph=new graph2(numberOfVertices,NewVertices,NewVerticesName);
    NewGraph.setSearchLevel(1);
    NewGraph.CreateEdges();
    NewGraph.Iteration();
    NewGraph.FindNeighbours();
    NewGraph.Forecast();
    NewGraph.RankbyForecastValue();
    NewVertices=NewGraph.VertexList;
    try{
        boolean append=true;
        DataOutputStream Fout = new DataOutputStream(new
        FileOutputStream("output[1].dat", append));
        String aFileName="outputPath"+"["+1+"]"+" ".dat";
        Date time=new Date();
        DataOutputStream aFout=new DataOutputStream(new
        FileOutputStream(aFileName, append));
        String aFormat="\n##### THE SEARCHED
URL#####"+ "\n\nThe Base Url"+Baseurl+"\nGenerated at "+time.getDate()+
time.getHours()+time.getMinutes()+time.getSeconds();
        aFout.writeUTF(aFormat);
        for(int k=0;k<SearchInfo[0].size();k++){
            vertices2
atemp=(vertices2) SearchInfo[0].elementAt(k);

```

```

        String temp="\n"+"*****"+atemp.Name+"*****"+"The
forecast value is "+atemp.ForecastValue+"the relative value is
+atemp.RelativeValue1;
        byte[] tempbytes=temp.getBytes();
        aFout.write(tempbytes);
        if(atemp.Adjacency!=null){
            for(int
n=0;n<atemp.Adjacency.size();n++){
                String
temp3="\n"+atemp.Adjacency.elementAt(n);
                byte[] format3=temp3.getBytes();
                aFout.write(format3);
            }
        }
        String format="\n##### THE FINAL RESULTS
#####"+list[0].size()+"*****"+NewVertices.length+"\n\n";
        Fout.writeUTF(format);
        for(int m=0;m<NewVertices.length;m++){
            String
temp2="\n"+"*****"+NewVertices[m].Name+"*****"+
            NewVertices[m].RcValue+"the forecast value is
            "+NewVertices[m].ForecastValue+NewVertices[m].visited+"
            vertex number"+ NewVertices.length+"\nthe hub
            "+NewVertices[m].Hub+"\n the
            authority"+NewVertices[m].Authority+"the forecast
            value"+NewVertices[m].RelativeValue1;//"the
            summary"+NewVertices[m].Pagesummary;
            byte[] format2=temp2.getBytes();
            Fout.write(format2);
            if(NewVertices[m].Adjacency!=null){
                for(int
n=0;n<NewVertices[m].Adjacency.size();n++){
                    String
temp3="\n"+NewVertices[m].Adjacency.elementAt(n);//
                    byte[] format3=temp3.getBytes();
                    Fout.write(format3);
                }
            }
        }
        Fout.close();
    }
    catch(IOException except){
        System.out.println("error occurs when output to
file");
    }
    Date Ctime1=new Date();
    String aTFormat1="\n#####The End Time of This
computation

```

```

#####"+ "\n"+Ctime1.getDate()+" "+Ctime1.getHours()+
        "+Ctime1.getMinutes()+" "+Ctime1.getSeconds()+
        "\n";
        FTout.writeUTF(aTFormat1);
        FTout.close();

    }
    catch(IOException excpt){
        System.out.println("error occurs when output to file");
    }
}

/* this function carries out a continuous search */
public vertices2[] NextLayerSearch(vertices2[] vertices,int i){
    VertexList.clear();
    int Level=i+1;
    Vector NotVisitedVertex=new Vector();
    for(int j=0;j<vertices.length;j++){
        String s=vertices[j].Name;
        Vector[] NoRepeatcountIndex=new Vector[termnumber];
        for(int M=0;M<termnumber;M++){
            NoRepeatcountIndex[M]=new Vector();
        }
        Vector NoRepeatUrlIndex=new Vector();
        if(vertices[j].visited==false){
            Vector Temp=new Vector();
            String content = pagescan1.getcontent(s);
            int checklink=content.indexOf("<a href=");
            if(checklink!=-1)
                continue;
            for(int k=0;k<termnumber;k++){
                pagescan1.wordcount[k].clear();
                pagescan1.wordurl[k].clear();
            }

            pagescan1.NoRepeatCountIndex[k].clear();
            pagescan1.wordurlIndex[k].clear();
            pagescan1.NewWordUrlIndex[k].clear();

        }
        pagescan1.wordpage.clear();
        pagescan1.NoRepeatUrlIndex.clear();
        pagescan1.newHits.clear();
        String[] linkurls=pagescan1.gethits(content);
        linkurls=pagescan1.Exacthits(linkurls);
        linkurls=pagescan1.NoRepeat(linkurls);
        pagescan1.countpage();
        Vector pagecollection=new Vector();
        pagecollection=pagescan1.wordpage;
        pagenumber=linkurls.length;
        termcount=pagescan1.wordcount;
        NoRepeatUrlIndex=pagescan1.NoRepeatUrlIndex;

        NoRepeatcountIndex=pagescan1.NoRepeatCountIndex;

```

```

        termfrequency=new
int [termnumber] [pagenumber];
        if (pagenumber!=0) {
        try{
            boolean append=true;
            int index=i+1;
            String
FileName="outputFrequency"+"["+index+"]"+" .dat";
            DataOutputStream Fout = new
DataOutputStream(new
                    FileOutputStream(FileName,append));
            String format="\n##### THE
FREQUENCY STATISTICS #####"+ "\n\n";
            Fout.writeUTF(format);
            for(int il=0;il<termnumber;il++){
                String
temp5="\n*****"+s+"*****";
                String temp3="\n NO REPEAT COUNT INDEX

                "+NoRepeatcountIndex[il]+"====="+termcount[il]+"linkurls"
                +linkurls.length+"count
                size"+NoRepeatcountIndex[il].size();
                byte[] format5=temp5.getBytes();
                byte[] format3=temp3.getBytes();
                Fout.write(format5);
                Fout.write(format3);
                String temp4="\n"+"NO REPEAT URL
INDEX "+NoRepeatUrlIndex+"---- -----";
                byte[] format4=temp4.getBytes();
                Fout.write(format4);
            }
            Fout.close();
        }
        catch(IOException excpt){
            System.out.println("error occurs when
output to file");
        }
        for(int il=0;il<termnumber;il++){
            int temp=termcount[il].size();
            for(int k=0;k<pagenumber;k++){
                for(int L=0; L<temp;L++){
                    System.out.println("the
pagenumber is "+pagenumber );
                    System.out.println("the No
repeat url index size is"+NoRepeatUrlIndex.size());
                    Integer
temp1=(Integer)NoRepeatcountIndex[il].elementAt(L);
                    Integer
temp2=(Integer)NoRepeatUrlIndex.elementAt(k);

                    if(temp2.intValue()==temp1.intValue()){
                        Integer
temp3=(Integer)termcount[il].elementAt(L);
                        termfrequency[il][k]=temp3.intValue();

```

```

                break;
            }
            else{

termfrequency[i1][k]=0;

            }

        }

        System.out.println("wordcount element:
"+pagescan1.wordcount[i1]);

    }

    try{
        int index=i+1;
        String
FileName="FrequencyMatrix"+"["+index+"]"+" .dat";
        boolean append=true;
        DataOutputStream Fout = new
DataOutputStream(new

        FileOutputStream(FileName, append));
        String format="\n##### THE MATRIX FREQUENCY STATISTICS

#####"+ "\n\n";
        Fout.writeUTF(format);
        for(int k=0;k<pagenumber;k++){
            for(int i1=0;i1<termnumber;i1++){
                String temp6="\n"+"the term
frequency is"+termfrequency[i1][k]+"==";
                byte[] format6=temp6.getBytes();
                Fout.write(format6);

            }

        }
        Fout.close();
    }
    catch(IOException excpt){
        System.out.println("error occurs when
output to file");
    }

    String[] Nextsummary=pagescan1.excerpts;
    vertices[j].setN(pagecollection);
    vertices[j].setPageNumber(pagenumber);
    vertices[j].setLinkTermFrequency(termfrequency);
    vertices[j].setFrequency();
    vertices[j].setLinkSummary(Nextsummary);

    for(int k=0;k<linkurls.length;k++){
        Temp.addElement(linkurls[k]);
    }
    vertices[j].setLinks(Temp);
    SearchInfo[i+1].addElement(vertices[j]);
    VertexList.addElement(vertices[j]);

```

```

        // If the page contained any links
        if ( linkurls != null )
        {
            for ( int k = 0; k < linkurls.length;
k++ )
            {
                if ( linkurls[ k ] != null
)
                list[i+1].addElement(linkurls[ k ]) ;
            }

            System.out.println( s );
        }
    }
}
if ( list[i+1].size() > MAX_LINKS )
    break;
}
/* here we begin to cut the left-hand nodes */
Vector[] newAdjacency=new Vector[SearchInfo[Level-1].size()];
for (int n=0;n<SearchInfo[Level-1].size();n++){
    newAdjacency[n]=new Vector();
    vertices2 tempVertex1=(vertices2) SearchInfo[Level-
1].elementAt(n);
    Vector TempVector1=tempVertex1.Adjacency;
    for (int k=0;k<tempVertex1.Adjacency.size();k++){
        String tempString=(String)TempVector1.elementAt(k);
        //vertices2 tempVector2=
        for(int l=0;l<SearchInfo[Level].size();l++){
            vertices2
tempVertex2=(vertices2) SearchInfo[Level].elementAt(l);
            if
(tempString.equalsIgnoreCase(tempVertex2.Name)){
                newAdjacency[n].addElement(tempString);
            }
        }
    }
    tempVertex1.Adjacency=newAdjacency[n];
}

System.out.println("here we begin to cut right hand nodes");
/*here we beging to cut the right-hand nodes */
/*find the top 6 nodes */
Vector AlladjacentLink=new Vector();
Vector ESvalues=new Vector();
double alpha=0.2;
for(int n=0;n<SearchInfo[Level].size();n++){
    vertices2
tempVertex1=(vertices2) SearchInfo[Level].elementAt(n);
    Vector TempVector=tempVertex1.Adjacency;
    for(int k=0;k<tempVertex1.Adjacency.size();k++){
        String tempString=(String)TempVector.elementAt(k);
        AlladjacentLink.addElement(tempString);
    }
}

```

```

        double
ForecastingValue=alpha*tempVertex1.ForecastValue+
        (1-alpha)*tempVertex1.RelativeValue1;
        ESvalues.addElement(new Double(ForecastingValue));
    }
}
System.out.println(ESvalues.size());
System.out.println(AlladjacentLink.size());
Vector RAlladjacentLink=new Vector();
//rank these links
for (int L = 0; L < AlladjacentLink.size(); L++) {

    int max = L;
    Double ES1=(Double)ESvalues.elementAt(L);
    for (int j = L+1 ; j <AlladjacentLink.size(); j++){
    Double ES2=(Double)ESvalues.elementAt(j);
        double R1=ES1.doubleValue();
        double R2=ES2.doubleValue();
        System.out.println(R1);
        System.out.println(R2);
        //double
R2=(VertexList[max].Authority*Aweight+VertexList[max].Hub*Hweight)/sum;

        if (R2> R1) {
            max = j;
        }

    }

}

String
temp1=(String)AlladjacentLink.elementAt(max);
    RAlladjacentLink.addElement(temp1);
    /* String temp2= (String)AlladjacentLink.elementAt(L);

        String Temp =temp1;

        temp1=temp2;
        temp2 =Temp;
        */
}
System.out.println(RAlladjacentLink.size());
System.out.println(SearchInfo[Level].size());
//extract top 6 nodes
Vector top6=new Vector();
if(RAlladjacentLink.size(>40){
    for(int k=0;k<40;k++){
        String tempstring=(String)AlladjacentLink.elementAt(k);
        top6.addElement(tempstring);
    }
}
else{
    for(int k=0;k<RAlladjacentLink.size();k++){
        String tempstring=(String)AlladjacentLink.elementAt(k);
        top6.addElement(tempstring);
    }
}

```

```

    }
    System.out.println(top6.size());
    Vector[] RnewAdjacency=new Vector[SearchInfo[Level].size()];
    for (int n=0;n<SearchInfo[Level].size();n++){
        RnewAdjacency[n]=new Vector();
        vertices2
tempVertex1=(vertices2)SearchInfo[Level].elementAt(n);
        Vector TempVector1=tempVertex1.Adjacency;
        for (int k=0;k<tempVertex1.Adjacency.size();k++){

            String tempString=(String)TempVector1.elementAt(k);
            //vertices2 tempVector2=
            for(int l=0;l<top6.size();l++){
                String tempstring2=(String)top6.elementAt(l);
                if
(tempString.equalsIgnoreCase(tempstring2)){

                    RnewAdjacency[n].addElement(tempString);
                    break;

                }

            }
            tempVertex1.Adjacency=RnewAdjacency[n];
        }

        // the following is to implement the sliding win
        for(int k=Level-1;k<Level+1;k++){
            for(int M=0;M<SearchInfo[k].size();M++){

                VertexList.addElement(SearchInfo[k].elementAt(M));

            }

            URLvertex=new vertices2[VertexList.size()];
            VertexList.copyInto(URLvertex);
            try{
                boolean append=true;
                DataOutputStream Fout = new DataOutputStream(new

                FileOutputStream("outputURLvertex.dat",append));
                String aFormat="\n##### THE URL
VERTEX#####"+"the URL vertex

                number"+URLvertex.length;;

                Fout.writeUTF(aFormat);
                for(int m=0;m<URLvertex.length;m++){
                    for(int j=0;j<URLvertex[m].termnumber;j++){
                        int[] frequency=URLvertex[m].Frequency;
                        String temp="\nthe processed URL
frequency"+frequency[j]+"the vertex

                        name"+URLvertex[m].Name;//URLvertex[i].Frequency[j];
                        byte[] format=temp.getBytes();
                        Fout.write(format);
                        for (int k=0;k<URLvertex[m].pagenumber;k++){

```

```

int [][]
linkfrequency=URLvertex[m].LinkTermFrequency;
String temp2="\nthe link
termfrequency"+linkfrequency[j][k];
byte[] format2=temp2.getBytes();
Fout.write(format2);
}
}
}
Fout.close();
}
catch(IOException excpt){
System.out.println("error occurs when output to
file"+excpt);
}
try{
boolean Tappend=true;
String TFileName="TotalComputingTime"+"["+i+"]"+" .txt";
DataOutputStream FTout = new DataOutputStream(new
FileOutputStream(TFileName, Tappend));
Date Ctime=new Date();
String aTFormat="\n#####The Start Time of This
computation
#####"+ "\n"+Ctime.getDate()+":"+Ctime.getHours()+
": "+Ctime.getMinutes()+":"+Ctime.getSeconds()+"\n"
+System.currentTimeMillis();
FTout.writeUTF(aTFormat);
Set[i]=new BaseSet2(URLvertex, termnumber);
Set[i].SetNewSearchUrl(list[i+1]);
Set[i].CreateLevelVertex();
Set[i].SetSearchLevel(i+2);
Set[i].CreateSet();
NewVertices=Set[i].Vertex;
int numberOfVertices=NewVertices.length;
try{
boolean append=true;
DataOutputStream Fout = new DataOutputStream(new
FileOutputStream("outputBaseset.dat", append));
String aFormat="\n##### THE URL
VERTEX#####"+
"\n\nThe Base Url";
Fout.writeUTF(aFormat);
for(int m=0; m<numberOfVertices; m++){
for(int j=0; j<termnumber; j++){
int []
frequency=NewVertices[m].Frequency;
String temp="\nthe processed URL
frequency "+frequency[j]+
"the vertex name"+NewVertices[m].Name;
byte[] format=temp.getBytes();
Fout.write(format);

```

```

        }
    }
    Fout.close();
}
catch(IOException excpt){
    System.out.println("error occurs when output to
file"+excpt);
}
this.size=pagescan1.size;
TFIDF[i]=new tfidf2(numberOfVertices,termnumber, weight,
NewVertices);
    TFIDF[i].SetSearchLevel(i+2);
    TFIDF[i].Normalization();
    //NewVertices=TFIDF[i].rankByRc();
    TFIDF[i].writeout();
    String[] NewVerticesName=Set[i].VerticesName;
    //int numberOfVertices=NewVertices.length;
    NewGraph[i]=new
graph2(numberOfVertices,NewVertices,NewVerticesName);
    NewGraph[i].setNewVertex(Set[i].NewVertex);
    NewGraph[i].setSearchLevel(i+2);
    NewGraph[i].CreateEdges();
    NewGraph[i].Iteration();
    NewGraph[i].FindNeighbours();
    NewGraph[i].Forecast();
    //NewGraph[i].RankbyrelativeValue1();
    NewGraph[i].RankbyForecastValue();
    NewVertices=NewGraph[i].VertexList;
    try{
        boolean append=true;
        int depthindex=i+2;
        String FileName="output"+"["+depthindex+"]"+" .dat";
        String
aFileName="outputPath"+"["+depthindex+"]"+" .dat";
        Date time=new Date();
        DataOutputStream aFout=new DataOutputStream(new
        FileOutputStream(aFileName, append));
        DataOutputStream Fout = new DataOutputStream(new
        FileOutputStream(FileName, append));
        String aFormat="\n##### THE SEARCHED
URL#####"\nGenerated at "+time.getDate()+" "+time.getHours()+" "+
        time.getMinutes()+" "+time.getSeconds();
        aFout.writeUTF(aFormat);
        for(int k=0;k<SearchInfo[i+1].size();k++){
            vertices2
atemp=(vertices2) SearchInfo[i+1].elementAt(k);
            String temp="\n"+"*****"+atemp.Name+"*****"+
the forecast value is "+atemp.ForecastValue+"the relative value
is"+atemp.RelativeValue1;
            byte[] tempbytes=temp.getBytes();
            aFout.write(tempbytes);
            if(atemp.Adjacency!=null){
                for(int
n=0;n<atemp.Adjacency.size();n++){
                    String
temp3="\n"+atemp.Adjacency.elementAt(n);

```

```

byte[] format3=temp3.getBytes();
aFout.write(format3);

    }

}

String format="\n##### THE FINAL RESULTS
##### "+list[i+1].size()+"*****"+NewVertices.length+"the size
of the stream "+size+"\n\n";
Fout.writeUTF(format);
for(int m=0;m<NewVertices.length;m++){
    String
temp2="\n"+"*****"+NewVertices[m].Name+"*****"+"
the Rc Value"+NewVertices[m].RcValue+"
"+NewVertices[m].visited+NewVertices[m].RelativeValueTrack+
"\nthe Forecast
Value"+NewVertices[m].ForecastValue+"\nthe
hub"+NewVertices[m].Hub+"the
authority"+NewVertices[m].Authority+
"the relative value is "+ NewVertices[m].RelativeValue1;
    byte[] format2=temp2.getBytes();
    Fout.write(format2);
    if(NewVertices[m].Adjacency!=null){
        for(int
n=0;n<NewVertices[m].Adjacency.size();n++){
            String
temp3="\n"+NewVertices[m].Adjacency.elementAt(n);
            byte[] format3=temp3.getBytes();
            Fout.write(format3);
        }
    }
}

Fout.close();
}

catch(IOException excpt){
    System.out.println("error occurs when output to
file");
}

Date Ctime1=new Date();
String aTFormat1="\n#####The End Time of This computation
#####"+"\n"+Ctime1.getDate()+":"+Ctime1.getHours()+":"+
+Ctime1.getMinutes()+":"+Ctime1.getSeconds()+
"\n"+System.currentTimeMillis();
FTout.writeUTF(aTFormat1);
FTout.close();
}

catch(IOException excpt){
    System.out.println("error occurs when output to file");
}
}

```

```

        return NewVertices;
    }
}

/* this class is used to scan the webpage*/

class Scanner
{
    private ReadNumbers1 ReadWeight;
    public double[] queryweight;
    public int Collection;
    public Vector wordpage;
    public Vector[] wordurl;
    public Vector[] NewWordUrlIndex;
    public Vector newHits;
    public String[] keywords;
    public String baseurl;
    public String foundurls[];
    public Integer[] URLcounts;
    public String hits[];
    public String excerpts[];
    public int[][] termfrequency;
    public Vector[] wordcount;
    public Vector[] validwordcount;
    public int size;
    public Vector[] wordurlIndex;
    public Vector[] NoRepeatCountIndex;
    public Vector NoRepeatUrlIndex;
    //private Vector[] wordcountIndex;
    private static final String AGENT_NAME="java-yahoo-search";
    private static final String AGENT_VERSION="1.0";
    private static String
SEARCH_URL="http://google.yahoo.com/bin/search?";
    private static int MAXIMUMPAGES=2;

    public Scanner( String[] searchwords ) throws IOException
    {
        this.keywords=searchwords;
        this.size=0;
        wordcount=new Vector[searchwords.length];
        wordpage=new Vector();
        NoRepeatUrlIndex=new Vector();

        ReadWeight=new ReadNumbers1(searchwords.length);
        this.queryweight=ReadWeight.weightassign();
        wordurl=new Vector[searchwords.length];
        wordurlIndex=new Vector[searchwords.length];
        NoRepeatCountIndex=new Vector[searchwords.length];
        NewWordUrlIndex=new Vector[searchwords.length];
        for(int k=0;k<searchwords.length;k++){
            wordurl[k]=new Vector();
            wordcount[k]=new Vector();
            wordurlIndex[k]=new Vector();
            NewWordUrlIndex[k]=new Vector();
        }
        String hitdata1="";

```

```

String[] hitdata2;
String querywords="";
for(int j=0;j<keywords.length;j++)
{
    querywords=keywords[j]+" "+querywords;
}
for(int i=0;i<MAXIMUMPAGES;i++)
{
    this.baseurl=createQuery(querywords,i);
    hitdata1=hitdata1+getContent(baseurl);
}
hitdata2=gethits(hitdata1);
hitdata2=Exacthits(hitdata2);
foundurls=NoRepeat(hitdata2);
foundurls=ValidUrl(foundurls);
this.countpage();
Collection=foundurls.length;
}
protected String createQuery(String searchTokens,int p){

    String
encodedSearchString=URLEncoder.encode(searchTokens.toString());

    //return "what=web&fmt=c&pg=q&q="+encodedSearchString+"&stq=";

SEARCH_URL=SEARCH_URL+"p="+encodedSearchString+"&b="+p*20+"&h=s";
    return SEARCH_URL;

}

public String getContent(String baseurl)
{
    String nextline;
    StringBuffer content=new StringBuffer();
try
{ // Create a URL from the base
    URL base = new URL( baseurl );
    URLConnection agent=base.openConnection();
    DataInputStream in=new
DataInputStream(agent.getInputStream());
    while((nextline=in.readLine())!=null)
    {
        String line = nextline.trim();
        if(!line.equals(""))
            content.append(nextline+"\n");
    }
    in.close();
}

catch( MalformedURLException e )
{
    System.out.println( "Invalid url " + baseurl );
}
catch(IOException excpt)
{
    System.out.println("failed I/O:"+excpt);
}
}

```

```

    }
        return content.toString();
    }

/* extract hits in this page */
public String[] gethits( String page)
{
    Vector v = new Vector();
    size=size+page.length();
    int c=0;
    int first=0;

    int last=page.lastIndexOf("<a href=");

    while(true)
    {
        first=page.indexOf("<a href=",first)+"<a href=".length();
        v.addElement(new Integer(first));
        if (first>=last)
            break;
    }
    Integer index[]=new Integer[v.size()];
    // Integer index[]=new Integer[v.toArray()];

    v.copyInto(index);
    String hitsection[]=new String[v.size()];
    if(index.length>1){
        for (int i=0;i<index.length;i++)
        {
            int A=index[i].intValue();
            if (i==(index.length-1))
                break;
            int B=index[i+1].intValue();
            //A=A+1;
            c=i;
            hitsection[i]=page.substring(A,B);
            System.out.println( i + "th"+"hitsection
is"+hitsection[i]);
        }
        System.out.println( "the c+1 is  "+c);

        System.out.println( "the index length is
"+index.length);
        System.out.println( "the last is  "+last);
        System.out.println( "the page length is  "+
page.length());
        hitsection[c+1]=page.substring(last,page.length());
        //System.out.println( c+1 + "th"+"hitsection
is"+hitsection[c+1]);
    }
    else{
        hitsection[index.length-
1]=page.substring(last,page.length());
    }
    return (hitsection);
}

```

```

public String[] Exacthits(String[] hitsection)
{
    int count=0;
    Vector D=new Vector();
    Vector v2= new Vector();
    Vector[] wordcountIndex=new Vector[keywords.length];
    for (int k=0;k<keywords.length;k++)
    {
        wordcountIndex[k]=new Vector();
        for (int j=0;j<hitsection.length;j++)
        {
            int wordfrequency=0;
            int flag=0;
            int L=keywords[k].length();
            for (int i=L;i<hitsection[j].length();i++)
            {
                if (hitsection[j].substring(i-
L,i).equalsIgnoreCase(keywords[k]))
                {
                    wordfrequency++;
                    System.out.println("wordfrequency"+
wordfrequency);
                }
                if(wordfrequency>0){
                    flag=1;
                }
            }
            if(flag==1)
            {
                count++;
                wordurlIndex[k].addElement(new Integer(j));
                wordcountIndex[k].addElement(new
Integer(wordfrequency));
                D.addElement(new Integer(j));
            }
            System.out.println("useful urls:"+count);
        }
    }
    //Vector v1=new Vector();
    //v1=NoRepeat(D);
    URLcounts=new Integer[D.size()];
    D.copyInto(URLcounts);
    hits=new String[D.size()];
    String tempstring="";
    String foundurls[] = new String[ D.size() ];
    //String excerpt[]=new String[D.size];
    for(int j=0;j<keywords.length;j++){
        for(int k=0;k<URLcounts.length;k++){
            {

```

```

//Integer
value=(Integer)wordurlIndex.get (keywords[j]);
hits[k]=hitsection[URLcounts[k].intValue()];
//System.out.println( k + "th"+"Exacthit
is"+hits[k]);

int from = hits[k].indexOf( "\"" );
int to = hits[k].indexOf( "\"", from + 1 );
if ( to > 0 )
tempstring = hits[k].substring( from + 1, to );
for(int m=0;m<wordurlIndex[j].size();m++){
Integer
value=(Integer)wordurlIndex[j].elementAt (m);
int temp=value.intValue();
System.out.println("the word count size is
"+wordcountIndex[j].size());
System.out.println("the word url size is
"+wordurlIndex[j].size());
if(temp==URLcounts[k].intValue()){
wordurl[j].addElement (new
String(""+tempstring));
System.out.println("the word url size
is
"+wordurlIndex[j].size());

wordcount [j].addElement (wordcountIndex[j].elementAt (m));
NewWordUrlIndex [j].addElement (wordurlIndex[j].elementAt (m));
}
}
v2.addElement( new String( "" + tempstring ) );
System.out.println( "Found " + tempstring );
foundurls[k]=tempstring;

String refinestring="";
int startindex=0;
int endindex=0;
int finalindex=hits[k].lastIndexOf(">");
String s=hits[k];
int n=0;
while(true)
{

startindex=s.indexOf("<",startindex+1);
endindex=s.indexOf(">",endindex+1);
if (endindex<startindex)
{
String
s2=s.substring(endindex+1,startindex);

String s3=s2.trim();
if(!s3.equalsIgnoreCase(""));
refinestring=refinestring+s3;
}
if(endindex>=finalindex) break;

hits[k]=foundurls[k].concat ("\n").concat (refinestring);

```

```

                System.out.println( "hits " + hits[k] );
            }
        }
    }
    return foundurls;
}

/* remove the repeated URLs */

public String[] NoRepeat(String[] Url, int index)
{
    Vector newUrl=new Vector();
    NoRepeatCountIndex[index]=new Vector();
    Vector removeIndex= new Vector();
    boolean unique=true;
    Vector newWordcount=new Vector();
    for(int i=0;i<Url.length;i++)
    {
        for(int j=0;j<newUrl.size();j++)
        {
            if(Url[i].equalsIgnoreCase((String)newUrl.elementAt(j)))
            {
                removeIndex.addElement(new Integer(i));
                unique=false;
                break;
            }
            else
                unique=true;
        }

        if(unique==true)
        {
            newUrl.addElement(Url[i]);
            System.out.println("the wordurlIndex[index] size is
            "+wordurlIndex[index].size());
            System.out.println("the url length is
            "+Url.length);

            NoRepeatCountIndex[index].addElement(NewWordUrlIndex[index].elementAt
            (i));

            newWordcount.addElement(wordcount[index].elementAt(i));
        }
        System.out.println("the remove index length"+
        removeIndex.size());
        System.out.println("the wordcount index length"+
        wordcount[index].size());
        int startIndex=0;
        for(int i=0;i<removeIndex.size();i++)
        {
            Integer temp=(Integer)removeIndex.elementAt(i);

            System.out.println(wordcount[index].elementAt(temp.intValue()));
            System.out.println("remove index: "+ temp.intValue()+" AT
            element "+i );
        }
    }
}

```

```

startIndex=((Integer)removeIndex.elementAt(i)).intValue()+1;

    }

    String [] newUrlArray=new String[newUrl.size()];
    for(int i=0;i<newUrl.size();i++)
    {
        newUrlArray[i]=(String)newUrl.elementAt(i);
    }
    wordcount[index]=new Vector(newWordcount);
    return newUrlArray;
}

public String[] NoRepeat(String[] Url)
{
    newHits=new Vector();
    Vector newUrl=new Vector();
    Vector removeIndex= new Vector();
    boolean unique=true;
    for(int i=0;i<Url.length;i++)
    {
        for(int j=0;j<newUrl.size();j++)
        {
            if(Url[i].equalsIgnoreCase((String)newUrl.elementAt(j)))
            {
                removeIndex.addElement(new Integer(i));
                unique=false;
                break;
            }
            else
                unique=true;
        }

        if(unique==true)
        {
            newUrl.addElement(Url[i]);
            newHits.addElement(hits[i]);
            NoRepeatUrlIndex.addElement(URLcounts[i]);
        }
    }
    int startIndex=0;
    for(int i=0;i<removeIndex.size();i++)
    {
        Integer temp=(Integer)removeIndex.elementAt(i);

startIndex=((Integer)removeIndex.elementAt(i)).intValue()+1;

    }
}

```

```

String [] newUrlArray=new String[newUrl.size()];
excerptts=new String[newHits.size()];
System.out.println("the hits size is "+newHits.size());
System.out.println("the urlsize is "+newUrl.size());
for(int i=0;i<newUrl.size();i++)
{
    newUrlArray[i]=(String)newUrl.elementAt(i);
    excerptts[i]=(String)newHits.elementAt(i);
}

return newUrlArray;
}

public void display(){
    try{
        boolean append=true;
        DataOutputStream Fout = new DataOutputStream(new
        FileOutputStream("output.dat",append));
        String format="##### this is the test scanner
search result
        #####"+ "\n\n";
        Fout.writeUTF(format);
        for (int i=0;i<foundurls.length;i++){
            for(int j=0;j<keywords.length;j++){
                System.out.println("the final urls are
"+foundurls[i]);
                System.out.println("the found url length is
"+foundurls.length);
                System.out.println("the wordcount size is
"+wordcount[j].size());
                Integer
tempinteger=(Integer)wordcount[j].elementAt(i);
                String
temp2="\n"+"*****"+foundurls[i]+"*****"+tempinteger.intValue();
                byte[] format2=temp2.getBytes();
                Fout.write(format2);
            }
        }
        for(int i=0;i<keywords.length;i++){
            String
temp3="\n"+"*****"+"wordcountsize"+wordcount[i].size()
+*****wordpagenumber"+wordpage.elementAt(i);
            byte[] format3=temp3.getBytes();
            Fout.write(format3);
        }
    }
    catch(IOException excpt){
        System.out.println("error occurs when output to file");
    }
    for (int i=0;i<foundurls.length;i++)
    {
        System.out.println("the final urls are "+foundurls[i]);
    }
}

```

```

        System.out.println("the total url is "+foundurls.length);
        for(int k=0;k<keywords.length;k++){
            System.out.println("the collection is "+ Collection);
            System.out.println("the wordurl is "+ wordurl[k].size());
            System.out.println("the wordpage is "+ wordpage.size());
            System.out.println("the wordpage is "+
wordpage.elementAt(k));
            System.out.println("the wordcount is "+
wordcount[k].size());
        }
    }

    /* count the term frequency of each link */

    public void countpage(){
        termfrequency=new int[keywords.length][foundurls.length];
        String termpage[];
        for(int i=0;i<keywords.length;i++){
            termpage=new String[wordurl[i].size()];
            wordurl[i].copyInto(termpage);
            wordpage.addElement(new Integer(NoRepeat(termpage,
i).length));
        }
    }

    public String[] ValidUrl(String[] Urls){

        for(int i=0; i<Urls.length;i++){
            int index=Urls[i].indexOf("**");
            if(index===-1){

                foundurls[i]=Urls[i];
            }
            else{

foundurls[i]=Urls[i].substring(index+1,Urls[i].length());
            }
        }
        for(int j=0;j<foundurls.length;j++){
            int index2=foundurls[j].indexOf("site");
            if((foundurls[j].startsWith("/bin/query"))&&(index2!=-1))
{

foundurls[j]=foundurls[j].substring(index2+5,foundurls[j].length());
            foundurls[j]="http://"+foundurls[j];
        }
        else {
            foundurls[j]=foundurls[j];
        }
    }
    return foundurls;
}
}

```

```

public class TestIndepthSearch10{
    public static void main(String[] args) throws IOException
    {

        try{
            boolean append=true;
            DataOutputStream Fout = new DataOutputStream(new
            FileOutputStream("StartTime.dat",append));
            Date time=new Date();
            String aFormat="\n#####The Start Time of This program
#####"+ "\n"

            +time.getDate()+":"+time.getHours()+":"
            +time.getMinutes()+":"+time.getSeconds();
            Fout.writeUTF(aFormat);
            Fout.close();
        }

        catch(IOException excpt){
            System.out.println("error occurs when output
to file");
        }

        indepthsearch11 indepth = new indepthsearch11(args);
        int SearchDepth=7;
        indepth.search();
        vertices2[] Vertex=indepth.NewVertices;
        int MAXLINK=500;
        Vector aLinkList=new Vector();
        for(int i=0;i<SearchDepth;i++){

            Vertex=indepth.NextLayerSearch(Vertex,i);
            System.out.println("this is the "+i +"th
nextlayersearch");
        }

        try{
            boolean append=true;
            DataOutputStream Fout = new DataOutputStream(new

            FileOutputStream("EndTime.dat", append));
            Date time=new Date();
            String aFormat="\n#####The End Time of This
program
#####"+ "\n"+time.getDate()+":"+time.getHours()+
            ":"+time.getMinutes()+":"+time.getSeconds();
            Fout.writeUTF(aFormat);
            Fout.close();
        }

        catch(IOException excpt){

```

```

        System.out.println("error occurs when output
to file");
    }
}

/* this simple class is used to assign weight for each query term */

import java.io.*; /*tell the java compiler that we'll be doing i/o */

public class ReadNumbers1 {
    private int weightnumber;
    private double[] weightarray;
    public ReadNumbers1(int weightnumber){
        this.weightnumber=weightnumber;
        this.weightarray=new double[weightnumber];
    }
    public double[] weightassign() throws IOException{

        BufferedReader inputStream = new BufferedReader
            (new InputStreamReader(System.in));
        for(int i=0;i<weightnumber;i++)
        {
            System.out.println("Enter a real number:");

            weightarray[i]=Double.valueOf(inputStream.readLine().trim()).doubleValue();
        }
        for(int i=0;i<weightnumber;i++)
        {
            System.out.print(weightarray[i]+",");
        }
        return weightarray;
    }
}

/* this class is used to construct a node for each URL */

import java.net.*;
import java.io.*;
import java.util.*;
import java.lang.*;

class vertices2{
    public String Name;
    public double ForecastValue;
    public double RelativeValue1;
    public double RelativeValue2;
    public double RcValue;
    public double Hub;
    public Vector RelativeValueTrack;
}

```

```

public double similarity;
public double Authority;
public Vector Adjacency;
//public Vector TempSim;
public boolean visited;
public String Pagesummary;
public String[] LinkSummary;
public double[] LinkRcValue;
public Vector N;
public Vector ForecastValueTrack;
public int termnumber;
public int pagenumber;
public int[] Frequency;
public int[][] LinkTermFrequency;
//public Vector Links;
public vertices2(String Name , Vector Adjacency ,int atermnumber){
    this.Name=Name;
    this.Adjacency=Adjacency;
    this.termnumber=atermnumber;
    this.visited=false;
    this.Hub=1;
    this.Authority=1;
    this.similarity=similarity;
    //this.TempSim=TempSim;
    this.RelativeValueTrack=new Vector();
    this.ForecastValueTrack=new Vector();

    this.ForecastValue=0.0;
    this.RelativeValue1=0.0;
    this.RelativeValue2=0.0;
    this.RcValue=0.0;
    this.Frequency=new int[termnumber];

}
public vertices2(){};
public String getName(){ return Name;}
public Vector getAdjacency(){return Adjacency;}
public void setPageNumber(int
apagenumber){this.pagenumber=apagenumber;};
public void setVisited(){this.visited=true;}
public void setHub(double hub){ this.Hub=hub;}
public void setLinkSummary(String[]
aLinkSummary){this.LinkSummary=aLinkSummary;};
public void setAuthority(double authority){
this.Authority=authority;}
public void setSimilarity(double
asimilarity){this.similarity=asimilarity;}
public void setSummary(String asummary ){this.Pagesummary=asummary;};
public void setRelativeValue1(double aRelativeValue)
{
    this.RelativeValue1=aRelativeValue;
    RelativeValueTrack.addElement(new Double(aRelativeValue));
}
public void setRelativeValue2(double RelativeValue)
{
    this.RelativeValue2=RelativeValue;
}

```

```

        RelativeValueTrack.addElement(new Double(RelativeValue));
    }
    public void setForecastValue(double ForecastValue)
    {
        this.ForecastValue=aForecastValue;
        this.ForecastValueTrack.addElement(new Double(aForecastValue));
    }
    public void setRcValue(double aRcValue){this.RcValue=aRcValue;}
    public void setLinkRcValue(double[] aLinkRcValue)
    {
        this.LinkRcValue=aLinkRcValue;
    }
    public void setN(Vector aN){this.N=aN;};
    public void setLinks(Vector atemp){this.Adjacency=atemp;};
    public void setLinkTermFrequency(int[][] aFrequency)
    {
        this.LinkTermFrequency=new int[termnumber][pagenumber];
        this.LinkTermFrequency=aFrequency;
    }
    public void setFrequency(){
        for (int i=0;i<termnumber;i++){
            int temp=0;
            for(int j=0;j<pagenumber;j++){
                temp=temp+LinkTermFrequency[i][j];
            }
            Frequency[i]=temp;
        }
    }
    public void setFrequency(int[] aFrequency){Frequency=aFrequency;};
}

```

/* this class is used to construct a graph set */

```

import java.io.*;
import java.net.*;
import java.lang.*;
import java.util.*;

class BaseSet2{

    public Vector UrlList;
    public vertices2[] Vertex;
    private vertices2[] Tempvertices;
    public String[] VerticesName;
    //public Vector Sim;
    public Vector HitSection;
    //public Vector NewSim;
    public Vector NewHitSection;
    public Vector NewVertex;
    public Vector VertexFrequency;
    public Vector NewVertexFrequency;
    public Vector LevelSearch;
    public int SearchIndex;
    public vertices2[] LevelVertex;
}

```

```

public vertices2[] PreviousVertex;
public int termnumber;
public BaseSet2(vertices2[] aTempvertices,int atermnumber){
    this.Tempvertices=aTempvertices;
    this.termnumber=atermnumber;
    UrlList=new Vector();
    this.NewVertex=new Vector();
    HitSection=new Vector();
    NewHitSection=new Vector();
    NewVertexFrequency=new Vector();
    VertexFrequency=new Vector();
}
public void SetNewSearchUrl(Vector NewUrl){this.LevelSearch=NewUrl;}
public void CreateLevelVertex(){
    this.LevelVertex=new vertices2[LevelSearch.size()];
    for(int i=0;i<LevelSearch.size();i++){
        String s=(String)LevelSearch.elementAt(i);
        LevelVertex[i]=new vertices2(s,null,termnumber);
    }
}
public void SetSearchLevel(int n){SearchIndex=n;}

public void SetPreviousVertex(vertices2[] aPreviousVertex){
    this.PreviousVertex=aPreviousVertex;
}
/* create a set for the current search level */
public vertices2[] CreateSet(){
    //Vector newHits=new Vector();
    Vector newUrl=new Vector();
    Vector removeIndex= new Vector();
    boolean unique=true;

    for(int i=0;i<Tempvertices.length;i++){

        System.out.println("the tempvertices length is
"+Tempvertices.length);
        UrlList.addElement(Tempvertices[i].Name);

        HitSection.addElement(Tempvertices[i].Pagesummary);
        Vector TempFrequencyVector=new Vector();
        for(int k=0;k<termnumber;k++){
            TempFrequencyVector.addElement(new

Integer(Tempvertices[i].Frequency[k]));
        }

        VertexFrequency.addElement(TempFrequencyVector);
        Vector TEMP=new Vector();
        TEMP=Tempvertices[i].Adjacency;
        Vector TEMP2=new Vector();
        int
TempLinkFrequency[][]=Tempvertices[i].LinkTermFrequency;
        for(int k2=0;k2<Tempvertices[i].pagenumber;k2++){
            Vector TempLinkFrequencyVector =new Vector();
            for(int k1=0;k1<termnumber;k1++){
                TempLinkFrequencyVector.addElement(new

```

```

Integer (Tempvertices[i].LinkTermFrequency[k1][k2]));
    }

VertexFrequency.addElement (TempLinkFrequencyVector);
    }
    for(int j=0;j<Tempvertices[i].Adjacency.size();j++){

        String[] tempsummary=Tempvertices[i].LinkSummary;

        UrlList.addElement (TEMP.elementAt(j));
        HitSection.addElement(tempsummary[j]);

    }
}
String[] Url=new String[UrlList.size()];
UrlList.copyInto(Url);
for(int k=0;k<Url.length;k++)
{
    for(int m=0;m<newUrl.size();m++)
    {
if(Url[k].equalsIgnoreCase((String)newUrl.elementAt(m))
        {
            removeIndex.addElement(new Integer(k));
            unique=false;
            break;
        }
        else
            unique=true;
    }

    if(unique==true)
    {
        newUrl.addElement(Url[k]);
        NewHitSection.addElement(HitSection.elementAt(k));

NewVertexFrequency.addElement (VertexFrequency.elementAt(k));
    }
}

VerticesName=new String[newUrl.size()];
Vertex=new vertices2[newUrl.size()];

{
    Vertex[i]=new vertices2("",null,termnumber);
    Vertex[i].setSummary((String)NewHitSection.elementAt(i));
    int[] tempfrequency=new int[termnumber];

    Vector
tempvector=(Vector)NewVertexFrequency.elementAt(i);

    for(int k=0;k<termnumber;k++){
        Integer tempvalue=(Integer)tempvector.elementAt(k);
        tempfrequency[k]=tempvalue.intValue();
    }
}

```

```

        Vertex[i].setFrequency(tempfrequency);
        VerticesName[i]=(String)newUrl.elementAt(i);
        Vertex[i].Name=VerticesName[i];
        //int tempindex=0;
        for(int j=0;j<Tempvertices.length;j++){

if (VerticesName[i].equalsIgnoreCase(Tempvertices[j].Name)) {
Vertex[i].Adjacency=Tempvertices[j].Adjacency;
                //Vertex[i].TempSim=Tempvertices[j].TempSim;

Vertex[i].similarity=Tempvertices[j].similarity;
Vertex[i].ForecastValueTrack=Tempvertices[j].ForecastValueTrack;
Vertex[i].setForecastValue(Tempvertices[j].ForecastValue);
Vertex[i].setLinkSummary(Tempvertices[j].LinkSummary);
Vertex[i].setFrequency(Tempvertices[j].Frequency);
Vertex[i].setLinkTermFrequency(Tempvertices[j].LinkTermFrequency);
Vertex[i].setPageNumber(Tempvertices[j].pagenumber);

                Vertex[i].visited=true;

                break;

        }

        //tempindex=j;
}
/*    if (tempindex==Tempvertices.length-1){
            Double Temp=(Double)NewSim.elementAt(i);
            double tempvalue=Temp.doubleValue();
            Vertex[i].similarity=tempvalue;
            NewVertex.addElement(Vertex[i]);
        }

        Double Temp=(Double)NewSim.elementAt(i);
        double tempvalue=Temp.doubleValue();
        Vertex[i].similarity=tempvalue;
        NewVertex.addElement(Vertex[i]);
        */
}

return Vertex;
}

```

```

public void FindNewVertex(){

    for(int i=0;i<LevelVertex.length;i++){
        int flag=0;
        for(int j=0;j<PreviousVertex.length;j++){

            if(LevelVertex[i].Name.equalsIgnoreCase(PreviousVertex[j].Name)){
                break;
            }
            flag=1;
        }
        if(flag==1){
            NewVertex.addElement(LevelVertex[i]);
        }
    }

    try{
        boolean append=true;
        String
        FileName="outputNewVertex"+"["+SearchIndex+"]"+" .dat";
        DataOutputStream Fout = new DataOutputStream(new
            FileOutputStream(FileName,append));
        //FileOutputStream Fout=new
        FileOutputStream("outputNewVertex",append);
        String format="##### THE Newvertex

        #####"+LevelVertex.length+"\n\n";
        Fout.writeUTF(format);
        for(int i=0;i<LevelVertex.length;i++){
            String temp="\nthe LevelVertex
name"+LevelVertex[i].Name+"\n";
            Fout.writeUTF(temp);
        }
        for(int i=0;i<NewVertex.size();i++){
            vertices2 temp=(vertices2)NewVertex.elementAt(i);
            String temp3="\n the new vertex size
"+NewVertex.size()+"the vertex
name"+temp.Name+" the
tempvertices "+Tempvertices.length
+"url list "+UrlList.size()+"the number of vertex
"+Vertex.length;

            byte[] format3=temp3.getBytes();
            Fout.write(format3);
        }
        Vector temp5=new Vector();
        /*
        for(int j=0;j<Tempvertices.length;j++){

            temp5=Tempvertices[j].Adjacency;
            for(int
k=0;k<Tempvertices[j].Adjacency.size();k++){
                String
temp6="\n"+(String)Tempvertices[j].Name+"\n"+"this is the "+k+"th
vertex"+(String)temp5.elementAt(k);
                byte[] format5=temp6.getBytes();

```

```

        Fout.write(format5);
    }
}
for(int i=0;i<newUrl.size();i++){
    String temp="\n"+"the vertexname "+VerticesName[i];
    byte[] format7=temp.getBytes();
    Fout.write(format7);
}
*/
Fout.close();
}

catch(IOException excpt){
    System.out.println("error occurs when output to file");
}
}
}

```

/* this class is used to construct a graph and compute the Hub value, Authority value, page value, forecasting value */

```

import java.net.*;
import java.io.*;
import java.util.*;
import java.lang.*;

class graph2{
    public vertices2 VertexList[];
    private int AdjMatrix[][];
    private int NumberOfVertex;
    private int maxVertex;
    public Vector NewVertex;
    private String[] vertexName;
    public Vector[] NeighbourVector;
    private double Aweight=0.9;
    private double Hweight=0.1;
    public double Hub;
    public double Authority;
    public static int IterationTimes=40;
    public int SearchLevel;

    public graph2(int nvert, vertices2[] aVertexList,String[]
avertexName) {
        this.VertexList=aVertexList;
        AdjMatrix = new int[nvert][nvert];
        NumberOfVertex = nvert;
        this.vertexName=avertexName;
        this.NewVertex=new Vector();
        // initialize adjacency matrix entries to 0
        for(int i=0; i<NumberOfVertex; i++){
            for (int j=0; j<NumberOfVertex; j++){
                AdjMatrix[i][j]=0;
            }
        }
    }
}

```

```

    }

    public void setNewVertex(Vector aNewVertex){
        NewVertex=aNewVertex;
        System.out.println("the new vertex length is "+NewVertex.size());
    }

    public void setSearchLevel(int
aSearchLevel){this.SearchLevel=aSearchLevel;};
    public void CreateEdges(){
        for (int i=0;i<NumberOfVertex;i++){
            Vector temp=new Vector();
            temp=VertexList[i].Adjacency;
            if(temp!=null){
                for(int j=0;j<temp.size();j++){
                    for(int k=0;k<NumberOfVertex;k++){
                        String
tempString=(String)temp.elementAt(j);
                        System.out.println("tempstring is "+
tempString);
                        System.out.println("the vertex name is
"+ vertexName[k]);
                        if
(tempString.equalsIgnoreCase(vertexName[k]))
                            AdjMatrix[i][k]=1;
                    }
                }
            }
        }
        try{
            boolean append=true;
            String
FileName="outputAdjacency"+"["+SearchLevel+"]"+" .txt";
            DataOutputStream Fout = new DataOutputStream(new
FileOutputStream(FileName,append));

            String format="##### THE Adjacencymatrix
Result #####"\n\n";
            Fout.writeUTF(format);

            for(int k=0;k<NumberOfVertex;k++){
                String temp2="";
                for(int i=0;i<NumberOfVertex;i++){
                    temp2=AdjMatrix[k][i]+temp2;
                }
                String Temp3="\n"+temp2;
                byte[] format2=Temp3.getBytes();
                Fout.write(format2);
            }
            Fout.close();
        }
        catch(IOException excpt){
            System.out.println("error occurs when output to file");
        }
    }
}

```

```

    }
    public void Iteration(){
        try{
            boolean append=true;
            String
FileName="GraphComputingTime"+"["+SearchLevel+"]"+" .txt";
            DataOutputStream Fout = new DataOutputStream(new
                FileOutputStream(FileName,append));
            Date time=new Date();

            String aFormat="\n#####The Start Time of
Iteration #####"+ "\n" +time.getDate()+":"+time.getHours()+":"+
                time.getMinutes()+":"+time.getSeconds()+
                +"\n"+System.currentTimeMillis();
            Fout.writeUTF(aFormat);
            RvValue=VertexList[i].Authority*Aweight*VertexList[i].RcValu
e
            +VertexList[i].Hub*Hweight*VertexList[i].RcValue;
            for(int m=0;m<IterationTimes;m++){
                double HubSquareSum=0;
                double AuthoritySquareSum=0;
                for(int i=0;i<NumberOfVertex;i++){
                    double tempAuthority=0;
                    double tempHub=0;

                    for(int j=0;j<NumberOfVertex;j++){
                        if(AdjMatrix[j][i]==1)
                            tempAuthority=VertexList[j].Hub+tempAuthority;
                        if(AdjMatrix[i][j]==1)
                            tempHub=VertexList[j].Authority+tempHub;
                    }
                    VertexList[i].setAuthority(tempAuthority);
                    VertexList[i].setHub(tempHub);

                HubSquareSum=Math.pow(VertexList[i].Hub,2)+HubSquareSum;

                AuthoritySquareSum=Math.pow(VertexList[i].Authority,2)
                    +AuthoritySquareSum;
                }
                HubSquareSum=Math.sqrt(HubSquareSum);
                AuthoritySquareSum=Math.sqrt(AuthoritySquareSum);
                for(int k=0;k<NumberOfVertex;k++){

                    double NewHub=VertexList[k].Hub/HubSquareSum;
                    double
NewAuthority=VertexList[k].Authority/AuthoritySquareSum;
                    VertexList[k].setAuthority(NewAuthority);
                    VertexList[k].setHub(NewHub);
                }
            }
            Date time1=new Date();
            String aFormat1="\n#####The end Time of Iteration
#####"+ "\n"
            +time1.getDate()+":"+time1.getHours()+":"+time1.getMinutes()+":"+
                +time1.getSeconds()+ "\n"+System.currentTimeMillis();

```

```

        Fout.writeUTF(aFormat1);
        Fout.close();
    }

    catch(IOException excpt){
        System.out.println("error occurs when output to
file");
    }

    double sum1=0.0;
    double sum2=0.0;
    double[] R=new double[NumberOfVertex];
    for(int i=0;i<NumberOfVertex;i++){
        try{
            boolean append =true;
            DataOutputStream os=new DataOutputStream(new
FileOutputStream("outputIterationRCValue.dat",append));
            String title="\n##### the RC value for each new
vertex
#####+"\n the Rc
value"+VertexList[i].RcValue+
            "\nthe hub"+VertexList[i].Hub+
            "the authority"+VertexList[i].Authority;
            byte[] format1=title.getBytes();
            os.write(format1);
        }
        catch(IOException excp){
            System.out.println("error when write to
file"+excp);
        }

        sum1=sum1+(VertexList[i].Authority*Aweight*VertexList[i].RcValue+
VertexList[i].Hub*Hweight*VertexList[i].RcValue);

        sum2=sum2+(VertexList[i].Authority*Aweight+VertexList[i].Hub*Hweight)
;
    }
    try{
        double meanvalue=sum1/NumberOfVertex;
        boolean append =true;
        DataOutputStream os=new DataOutputStream(new
FileOutputStream("outputsumValue.dat",append));
        String title="\n##### the sum value for each
level search #####+"\n"+sum1+"\n the mean value in this level
"+meanvalue;
        byte[] format1=title.getBytes();
        os.write(format1);
    }
    catch(IOException excp){
        System.out.println("error when write to
file"+excp);
    }
}

```

```

        for(int i=0;i<NumberOfVertex;i++){

            double
temp1=((VertexList[i].Authority*Aweight*VertexList[i].RcValue+
            VertexList[i].Hub*Hweight*VertexList[i].RcValue))/sum1;

            doubletemp2=((VertexList[i].Authority*Aweight+
                VertexList[i].Hub*Hweight))/sum2;
            VertexList[i].setRelativeValue1(temp1);
            VertexList[i].ForecastValue=temp1;

            VertexList[i].setRelativeValue1(temp1);
            try{
                boolean append=true;
                String
FileName="RvValue"+"["+SearchLevel+"]"+" .txt";
                DataOutputStream Fout = new DataOutputStream(new
                    FileOutputStream(FileName, append));
                Double
RvValue=VertexList[i].Authority*Aweight*VertexList[i].RcValue
                    +VertexList[i].Hub*Hweight*VertexList[i].RcValue;

                String temp="\n"+RvValue;
                byte[] format=temp.getBytes();
                Fout.write(format);

                Fout.close();
            }
            catch(IOException excpt){
                System.out.println("error occurs when output to
file");
            }
        }
    }

    /* Rank nodes based on their values */
    public void RankbyrelativeValue2(){

        for (int i = 0; i < NumberOfVertex-1; i++) {
            int max = i;

            for (int j = i+1 ; j <NumberOfVertex; j++){

                if (VertexList[i].RelativeValue2 >
VertexList[max].RelativeValue2){
                    max = j;
                }
            }
        }
    }

```

```

    }

    vertices2 Temp =new vertices2();
    VertexList[max]=Temp;
    VertexList[max] = VertexList[i];
    VertexList[i] =Temp;

}

try{
    boolean append=true;
    DataOutputStream Fout = new DataOutputStream(new
    FileOutputStream("outputbyrelativevalue1.dat",append));
    String format="##### THE RESULTS ORDERED BY
RELATIVE VALUE2
#####"+"\n\n";
    Fout.writeUTF(format);
    for(int k=0;k<NumberOfVertex;k++){
        String temp2="\n"+"****the HUB
"+VertexList[k].Hub+"****"+
        "the authority is "+VertexList[k].Authority+
        "the relative value is
+VertexList[k].RelativeValue2
        +"the ForecastValue is
"+VertexList[k].ForecastValue;
        byte[] format2=temp2.getBytes();
        Fout.write(format2);

    }

    Fout.close();
}

catch(IOException excpt){
    System.out.println("error occurs when output to file");
}

}

public void RankbyrelativeValue1(){

    for (int i = 0; i < NumberOfVertex-1; i++) {
        int max = i;

        for (int j = i+1 ; j <NumberOfVertex; j++){

            if (VertexList[j].RelativeValue1 >
VertexList[max].RelativeValue1){
                max = j;
            }

        }

        vertices2 Temp =VertexList[max];

```

```

VertexList[max] = VertexList[i];
VertexList[i] = (vertices2)Temp;

}
try{
    boolean append=true;
    DataOutputStream Fout = new DataOutputStream(new
FileOutputStream("outputbyrelativevalue1.dat", append));

    String format="##### THE RESULTS ORDERED BY
RELATIVE VALUE1 #####"+'\n\n';
    Fout.writeUTF(format);
    for(int k=0;k<NumberOfVertex;k++){
        String temp2="\n"+VertexList[k].Name+"\n****the HUB
"+VertexList[k].Hub+"*****"+"the authority is
"+VertexList[k].Authority+"the relative value is
        "+VertexList[k].RelativeValue1+"the
ForecastValue is "
        +VertexList[k].ForecastValue;
        byte[] format2=temp2.getBytes();
        Fout.write(format2);

    }

    Fout.close();
}

catch(IOException excpt){
    System.out.println("error occurs when output to file");
}

}

public void rankbySum(){
    for (int i = 0; i < NumberOfVertex-1; i++) {
        int max = i;
        for (int j = i+1 ; j <NumberOfVertex; j++){
            double
Sum1=Math.pow(VertexList[j].Authority,2)+
            Math.pow(VertexList[j].Hub,2);
            double Sum2=Math.pow(VertexList[max].Authority,2)+
            Math.pow(VertexList[max].Hub,2);
            if (Sum1 > Sum2) {
                max = j;
            }

        }

    }

    vertices2 Temp =VertexList[max];
    VertexList[max] = VertexList[i];

```

```

VertexList[i] =(vertices2)Temp;
}
try{
    boolean append=true;
    DataOutputStream Fout = new DataOutputStream(new
FileOutputStream("outputbySum.dat", append));
    String format="##### THE RESULTS ORDERED BY SUM
#####"+"\n\n";
    Fout.writeUTF (format);
    for(int k=0;k<NumberOfVertex;k++){
        //System.out.println("this is the vertex "+
URLvertex[m].Name);
        String temp2="\n"+VertexList[k].Name+"\n****the HUB
"+VertexList[k].Hub+"*****"+ "the authority is
"+VertexList[k].Authority+"the relative value is
"+VertexList[k].RelativeValue2+"the
ForecastValue is
"+VertexList[k].ForecastValue;
        byte[] format2=temp2.getBytes();
        Fout.write(format2);
    }
    Fout.close();
}
catch(IOException excpt){
    System.out.println("error occurs when output to file");
}
}

/* Find neighbours of a new node */

public void FindNeighbours(){
    NeighbourVector=new Vector[NumberOfVertex];
    for(int i=0;i<NumberOfVertex;i++){
        NeighbourVector[i]=new Vector();
    }
    for(int i=0;i<NumberOfVertex;i++){
        for(int j=0;j<NumberOfVertex;j++){
            if (AdjMatrix[i][j]==1)
NeighbourVector[j].addElement (VertexList[i]);
        }
    }

    try{
        boolean append=true;
        String
FileName="outputbyNeighbour"+"["+SearchLevel+"]"+" .dat";

```

```

        DataOutputStream Fout = new DataOutputStream(new
FileOutputStream(FileName,append));
        String format="##### THE RESULTS OF NEIGHBOURS
#####"+ "\n\n";
        Fout.writeUTF(format);
        for(int i=0;i<NumberOfVertex;i++){
            String temp2="\n"+ "the vertex
name"+VertexList[i].Name+"\n";
            byte[] format2=temp2.getBytes();
            Fout.write(format2);
            for(int j=0;j<NeighbourVector[i].size();j++){
                vertices2
index=(vertices2)NeighbourVector[i].elementAt(j);
                String temp3="\n the neighbour name
"+index.Name+"the neighbour
                forecast
value"+index.ForecastValue+"\n";
                byte[] format3=temp3.getBytes();
                Fout.write(format3);
            }
        }
        Fout.close();
    }
    catch(IOException excpt){
        System.out.println("error occurs when output to file");
    }
}
/* Calculate the forecasting values of a node */

public void Forecast(){
    double Alpha=0.5;
    int MaxIndex=0;
    for(int i=0;i<NumberOfVertex; i++){
        double MaxValue=0.0;
        for(int j=0;j<NeighbourVector[i].size();j++){

            vertices2
temp=(vertices2)NeighbourVector[i].elementAt(j);
            if(temp.ForecastValue>=MaxValue)
                MaxValue=temp.ForecastValue;
                MaxIndex=j;
        }

        try{
            boolean append =true;
            String
FileName="outputMaxValue"+"["+SearchLevel+"]"+" .dat";
            DataOutputStream os=new DataOutputStream(new
FileOutputStream(FileName,append));
            String title="\n##### the maxvalue for each
vertex #####";

            byte[] format1=title.getBytes();
            os.write(format1);

```

```

String temp="\nthe max Value is "+MaxValue+"the
Relative
value"+VertexList[i].RelativeValue1;;
byte[] format2=temp.getBytes();
os.write(format2);
}
catch(IOException excp){
System.out.println("error when write to
file"+excp);
}
double Fvalue=Alpha*MaxValue+(1-
Alpha)*VertexList[i].RelativeValue1;
System.out.println("the new vertex
size"+NewVertex.size());
vertices2 temp2=(vertices2)VertexList[i];
temp2.setForecastValue(Fvalue);
}
}
public void RankbyForecastValue(){

for (int i = 0; i < NumberOfVertex-1; i++) {
int max = i;
for (int j = i+1 ; j <NumberOfVertex; j++){

R1=(VertexList[j].Hub*Hweight+VertexList[j].Authority*Aweight)/sum;
if (VertexList[j].ForecastValue >
VertexList[max].ForecastValue) {
max = j;
}

}

vertices2 Temp =VertexList[max];
VertexList[max] = VertexList[i];
VertexList[i] =(vertices2)Temp;

}
try{
boolean append=true;
String
FileName="outputbyForecastvalue"+"["+SearchLevel+"]"+" .dat";
DataOutputStream Fout = new DataOutputStream(new
FileOutputStream(FileName, append));
String format="##### THE RESULTS ORDERED BY
FORECAST VALUE
#####"+ "\n\n";
Fout.writeUTF(format);
for(int k=0;k<NumberOfVertex;k++){
String temp2="\n"+VertexList[k].Name+"\n****the HUB
"+VertexList[k].Hub+"*****"+ "the authority is
"+VertexList[k].Authority+"the relative value is
"+VertexList[k].RelativeValue1+"the ForecastValue
is
"+VertexList[k].ForecastValue+"the new vertex is
"+NewVertex.size());

```

```

        byte[] format2=temp2.getBytes();
        Fout.write(format2);

    }

    Fout.close();
}

catch(IOException excpt){
    System.out.println("error occurs when output to file");
}

}

public void RankbySim(){

    for (int i = 0; i < NumberOfVertex-1; i++) {
        int max = i;
        for (int j = i+1 ; j <NumberOfVertex; j++){
            if (VertexList[j].similarity >
VertexList[max].similarity) {
                max = j;
            }
        }

        vertices2 Temp =VertexList[max];
        VertexList[max] = VertexList[i];
        VertexList[i] =(vertices2)Temp;

    }
    try{
        boolean append=true;
        String
        FileName="outputbySimilarityValue"+"["+SearchLevel+"]"+" .dat";
        DataOutputStream Fout = new DataOutputStream(new

        FileOutputStream(FileName,append));
        String format="##### THE RESULTS ORDERED BY
SIMILARITY VALUE #####"+ "\n\n";
        Fout.writeUTF(format);
        for(int k=0;k<NumberOfVertex;k++){
            String temp2="\n"+VertexList[k].Name+"\n****the HUB
"+VertexList[k].Hub+"*****"+ "the authority is
"+VertexList[k].Authority+"the similarity value is
"+VertexList[k].similarity+"the new vertex is
"+NewVertex.size();
            byte[] format2=temp2.getBytes();
            Fout.write(format2);

        }

        Fout.close();

    }

    catch(IOException excpt){
        System.out.println("error occurs when output to file");
    }
}

```

```

    }
}
public void RankbyAuthority(){

    for (int i = 0; i < NumberOfVertex-1; i++) {
        int max = i;

        for (int j = i+1 ; j <NumberOfVertex; j++){

            //double
            R1=(VertexList[j].Hub*Hweight+VertexList[j].Authority*Aweight)/sum;
            //double
            R2=(VertexList[max].Authority*Aweight+VertexList[max].Hub*Hweight)/sum;
            if (VertexList[i].Authority >
VertexList[max].Authority) {
                max = j;
            }

        }

    }

    vertices2 Temp =new vertices2();
    VertexList[max]=Temp;
VertexList[max] = VertexList[i];
VertexList[i] =Temp;

}
try{
    boolean append=true;
    DataOutputStream Fout = new DataOutputStream(new
    FileOutputStream("outputbyAuthority.dat", append));
    //FileOutputStream Fout=new
FileOutputStream("output", append);
    String format="##### THE RESULTS ORDERED BY
AUTHORITY VALUE #####"\n\n";
    Fout.writeUTF(format);
    for(int k=0;k<NumberOfVertex;k++){
        //System.out.println("this is the vertex "+
URLvertex[m].Name);
        String temp2="\n"+VertexList[k].Name+"\n The
authority is
"+VertexList[k].Authority;
        byte[] format2=temp2.getBytes();
        Fout.write(format2);
    }

    Fout.close();
}
catch(IOException except){
    System.out.println("error occurs when output to file");
}
}

```

```

        //System.out.println("the hub after rank is
"+VertexList[k].Hub);

        //    System.out.println("the authority after rank is
"+VertexList[k].Authority);
    }

}

}

/* this class is used to compute the similarity value and RC value in the
first level search*/

import java.net.*;
import java.io.*;
import java.util.*;
import java.lang.*;

class tfidf{
    private int pagenumber;
    private int termnumber;
    public double sim[];
    private double Alpha=0.4;
    public double Rc[];
    public double MeanNorm;
    public double Norm[];
    public int termfrequency[][];
    private double queryweight[];
    public double[][] termweight;
    public String[] PageUrl;
    public String[] HitSection;
    private Vector N;
    private double sum[];
    public tfidf(int pagenumber, int termnumber, int[][]
wordcount,double[] queryweight,Vector N,String[] PageUrl,String[]
summary){
        this.PageUrl=PageUrl;
        this.HitSection=summary;
        this.pagenumber=pagenumber;
        this.termnumber=termnumber;
        this.termfrequency=wordcount;
        this.queryweight=queryweight;
        this.N=N;
        this.sum=new double[pagenumber];
        this.Rc=new double[pagenumber];
        this.Norm=new double[pagenumber];
        this.sim=new double[pagenumber];
        this.termweight=new double[termnumber][pagenumber];
        try{
            boolean append=true;
            DataOutputStream Fout = new DataOutputStream(new
                FileOutputStream("SeedUrl.dat",append)
            );

```

```

        String format="\n"+"##### THE SEED URLS
        #####"+'\n\n';
        Fout.writeUTF(format);
        for(int i=0;i<pagenumber;i++){
            String temp2="\n"+PageUrl[i]+"*****";
            byte[] format2=temp2.getBytes();
            Fout.write(format2);
        }
        Fout.close();
    }

    catch(IOException excpt){
        System.out.println("error occurs when output to
file");
    }
}

public void Normalization(){
    System.out.println("here begin the normalization
pagenumber"+pagenumber);
    System.out.println("here begin the normalization
termnumber"+termnumber);
    double tempweight[][]=new double[termnumber][pagenumber];

    for(int j=0;j<pagenumber;j++)
    {
        sum[j]=0;

        for (int i=0;i<termnumber;i++)
        {
            Integer temp=(Integer)N.elementAt(i);
            int temp2=temp.intValue();
            if(temp2==0){
                tempweight[i][j]=0.0;
            }
            else{
                double ratio=(double)pagenumber/temp2;
                tempweight[i][j]=(double)termfrequency[i][j]*(Math.log(r
atio)
                    *(1/Math.log(10))+1);
            }
            sum[j]=Math.pow(tempweight[i][j],2)+sum[j];
        }
    }
    double TempNormSum=0.0;
    for(int i=0;i<termnumber;i++){
        for(int j=0;j<pagenumber;j++){
            termweight[i][j]=tempweight[i][j]/Math.sqrt(sum[j]);
            Norm[j]=Math.sqrt(sum[j]);
            TempNormSum=TempNormSum+Math.sqrt(sum[j]);
        }
    }
}

```

```

    }
}
MeanNorm=TempNormSum/pagenunder;
double sumquery=0;
for(int i=0;i<termnumber;i++){

    sumquery=Math.pow(queryweight[i],2)+sumquery;
}
for(int i=0;i<termnumber;i++){
    queryweight[i]=queryweight[i]/Math.sqrt(sumquery);
}
for(int j=0; j<pagenunder;j++){
    sim[j]=0;
    for(int i=0;i<termnumber;i++){
        sim[j]=termweight[i][j]*queryweight[i]+sim[j];

    }

}
for(int j=0;j<pagenunder;j++){
    Rc[j]=Alpha*sim[j]+(1-Alpha)*Norm[j]/MeanNorm;
}
}
public String[] rankBySim(){
    System.out.println("the hit section size is
"+HitSection.length);
    System.out.println("the url size is "+PageUrl.length);
    System.out.println("the page number is "+pagenunder);

    for (int i = 0; i < pagenunder-1; i++) {
        int max = i;

        for (int j = i+1 ; j < pagenunder; j++){

            if (sim[j] > sim[max]) {
                max = j;
            }

        }

        double Temp = sim[max];
        sim[max] = sim[i];
        sim[i] = Temp;
        double Temp5=Rc[max];
        Rc[max]=Rc[i];
        Rc[i]=Temp5;
        String temp2=PageUrl[max];
        PageUrl[max]=PageUrl[i];
        PageUrl[i]=temp2;
        String temp3=HitSection[max];
        HitSection[max]=HitSection[i];
        HitSection[i]=temp3;
    }
}

```

```

        for(int k=0;k<termnumber;k++){
            int temp4=termfrequency[k][max];
            termfrequency[k][max]=termfrequency[k][i];
            termfrequency[k][i]=temp4;
        }
    }

    return PageUrl;
}

public String[] rankByRc(){
    System.out.println("the hit section size is
"+HitSection.length);
    System.out.println("the url size is "+PageUrl.length);
    System.out.println("the page number is "+pagenumber);
    for (int i = 0; i < pagenumber-1; i++) {
        int max = i;
        for (int j = i+1 ; j < pagenumber; j++){
            if (Rc[j] > Rc[max]) {
                max = j;
            }
        }
        double Temp = Rc[max];
        Rc[max] = Rc[i];
        Rc[i] = Temp;
        String temp2=PageUrl[max];
        PageUrl[max]=PageUrl[i];
        PageUrl[i]=temp2;
        double temp5=sim[max];
        sim[max]=sim[i];
        sim[i]=temp5;
        String temp3=HitSection[max];
        HitSection[max]=HitSection[i];
        HitSection[i]=temp3;
        for(int k=0;k<termnumber;k++){
            int temp4=termfrequency[k][max];
            termfrequency[k][max]=termfrequency[k][i];
            termfrequency[k][i]=temp4;
        }
    }

    return PageUrl;
}

public void writeout(){
    try{
        boolean append=true;
        DataOutputStream Fout = new DataOutputStream(new
FileOutputStream("outputbyTfidf1.dat", append));

        String format="\n"+"##### THE TFIDF
RESULTS#####"\n\n";

```

```

        Fout.writeUTF(format);
        for(int i=0;i<pagenumber;i++){
            String temp2="\n"+PageUrl[i]+"*****the
similarity"+sim[i]+"*****
            the Rc value "+Rc[i];
            byte[] format2=temp2.getBytes();
            Fout.write(format2);
            for(int j=0;j<termnumber;j++){
                String temp4="\n"+"the term frequency is
"+termfrequency[j][i]+"
                "+N.elementAt(j)+" "+pagenumber;
                byte[] format4=temp4.getBytes();
                Fout.write(format4);
            }
        }
        Fout.close();
    }
    catch(IOException excpt){
        System.out.println("error occurs when output to
file");
    }
}

/* this class is used to compute the similarity value and RC value for each
node*/

import java.net.*;
import java.io.*;
import java.util.*;
import java.lang.*;

class tfidf2{
    private int pagenumber;
    private int termnumber;
    public double sim[];
    public double SumRc;
    private double Alpha=0.4;
    public double Rc[];
    public double MeanNorm;
    public double Norm[];
    public int termfrequency[][];
    private double queryweight[];
    public double[][] termweight;
    public vertices2[] Page;
    public String[] HitSection;
    private int SearchLevel;
    private Vector N;
    private double sum[];
    public tfidf2(int pagenumber, int termnumber, double[]
        queryweight,vertices2[] Vertex){
        this.Page=Vertex;
        this.pagenumber=pagenumber;
        this.termnumber=termnumber;

```

```

this.queryweight=queryweight;
this.N=new Vector();
this.SumRc=0.0;
this.sum=new double[pagenumber];
this.Rc=new double[pagenumber];
this.Norm=new double[pagenumber];
this.sim=new double[pagenumber];
this.termweight=new double[termnumber][pagenumber];
try{
    boolean append=true;
    DataOutputStream Fout = new DataOutputStream(new
FileOutputStream("SeedUrl.dat", append));

        String format="\n"+"##### THE SEED URLS
#####"+ "\n\n";
        Fout.writeUTF(format);
        for(int i=0;i<pagenumber;i++){

            String temp2="\n"+Page[i].Name+"*****";
            byte[] format2=temp2.getBytes();
            Fout.write(format2);
        }
        Fout.close();
    }

    catch(IOException excpt){
        System.out.println("error occurs when output to
file");
    }
}

public void SetSearchLevel(int
aSearchLevel){this.SearchLevel=aSearchLevel;};
public void Normalization(){

    System.out.println("here begin the normalization
pagenumber"+pagenumber);
    System.out.println("here begin the normalization
termnumber"+termnumber);
    double tempweight[][]=new double[termnumber][pagenumber];

    for(int j=0;j<termnumber;j++)
    {
        int count=0;
        for (int i=0;i<pagenumber;i++)
        {
            if(Page[i].Frequency[j]>0)
                count++;
        }
        N.addElement(new Integer(count));
    }

    for(int j=0;j<pagenumber;j++)
    {

        sum[j]=0;

```

```

for (int i=0;i<termnumber;i++)
{
    Integer temp=(Integer)N.elementAt(i);
    int temp2=temp.intValue();
    if(temp2==0){
        tempweight[i][j]=0.0;
    }
    else{
        double ratio=(double)pagenumber/temp2;
tempweight[i][j]=(double)Page[j].Frequency[i]*(Math.log(
ratio)
*(1/Math.log(10))+1);
    }
    sum[j]=Math.pow(tempweight[i][j],2)+sum[j];
}
}
double TempNormSum=0.0;
for(int i=0;i<termnumber;i++){
    for(int j=0;j<pagenumber;j++){
termweight[i][j]=tempweight[i][j]/Math.sqrt(sum[j]);
        Norm[j]=Math.sqrt(sum[j]);
        TempNormSum=TempNormSum+Math.sqrt(sum[j]);
    }
}
MeanNorm=TempNormSum/pagenumber;
double sumquery=0;
for(int i=0;i<termnumber;i++){
    sumquery=Math.pow(queryweight[i],2)+sumquery;
}
for(int i=0;i<termnumber;i++){
    queryweight[i]=queryweight[i]/Math.sqrt(sumquery);
}
for(int j=0; j<pagenumber;j++){
    sim[j]=0;
    for(int i=0;i<termnumber;i++){
        sim[j]=termweight[i][j]*queryweight[i]+sim[j];
    }
    Page[j].setSimilarity(sim[j]);
}
for(int j=0;j<pagenumber;j++){
    Rc[j]=sim[j]*Norm[j]/MeanNorm;
    Page[j].setRcValue(Rc[j]);
    SumRc=SumRc+Rc[j];
}
}
public vertices2[] rankBySim(){
    System.out.println("the hit section size is
"+HitSection.length);
    System.out.println("the url size is "+Page.length);
    System.out.println("the page number is "+pagenumber);
}

```

```

        for (int i = 0; i < pagenumber-1; i++) {
            int max = i;

            for (int j = i+1 ; j < pagenumber; j++){
                if (sim[j] > sim[max]) {
                    max = j;
                }
            }

            double Temp = sim[max];
            sim[max] = sim[i];
            sim[i] = Temp;
            double Temp5=Rc[max];
            Rc[max]=Rc[i];
            Rc[i]=Temp5;
            vertices2 temp2=Page[max];
            Page[max]=Page[i];
            Page[i]=temp2;
        }

        return Page;
    }
    public vertices2[] rankByRc(){
        for (int i = 0; i < pagenumber-1; i++) {
            int max = i;

            for (int j = i+1 ; j < pagenumber; j++){

                if (Rc[j] > Rc[max]) {
                    max = j;
                }

            }

            double Temp = Rc[max];
            Rc[max] = Rc[i];
            Rc[i] = Temp;
            vertices2 temp2=Page[max];
            Page[max]=Page[i];
            Page[i]=temp2;
            double temp5=sim[max];
            sim[max]=sim[i];
            sim[i]=temp5;
        }

        return Page;
    }
    public void writeout(){
        try{
            boolean append=true;

```

```

        String
FileName="outputbyTFIDE"+"["+SearchLevel+"]"+" .dat";
        DataOutputStream Fout = new DataOutputStream(new
        FileOutputStream(FileName,append));
        String format="\n"+"##### THE TFIDE RESULTS

#####"+ "\n\n"+"the total RcValue"+SumRc;

        Fout.writeUTF(format);
        for(int i=0;i<pagenumber;i++){
            String temp2="\n"+Page[i].Name+"*****the
similarity"+sim[i]+"***** the Rc value "+Rc[i];
            byte[] format2=temp2.getBytes();
            Fout.write(format2);
            for(int j=0;j<termnumber;j++){
                String temp4="\n"+"the term frequency is
"+Page[i].Frequency[j];
                byte[] format4=temp4.getBytes();
                Fout.write(format4);
            }
        }
        Fout.close();
    }
    catch(IOException excpt){
        System.out.println("error occurs when output to
file");
    }
}
}

```

Appendix 2 The Searched URLs

A-1 “Automobile Manufacturer” Case (ES)

Level One

<http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1620>

0.4223452874385104

http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/

0.00965370328620864

http://dir.yahoo.com/Regional/Countries/Canada/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/

0.00826283875052485

<http://www.japanauto.com/>

0.00588277876957169

<http://www.aiam.org/>

0.00468378645210461

<http://yp.yahoo.com/py/ypResults.py?stx=manufacturer+automobile>

0.00239185455486392

Level Two:

<http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile>

0.49195917773042336

<http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile>

0.08212736856281715

<http://yp.search.yahoo.com/search/ypredirect?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile>

0.03938170883459406

http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D

0.0038421179350692104

http://story.news.yahoo.com/news?tmpl=story&u=/nm/20030415/pl_nm/environment_diesel_dc_2

0.001121058967534602

http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645

0.000821693780163709

Level Three

http://rd.yahoo.com/search/navbar/top/summary/manufacturer+automobile/*http://search.yahoo.com/search/dir?p=manufacturer+automobile

0.6045241277873227

http://rd.yahoo.com/search/navbar/regional/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&o=&g=&n=20&h=C&r=Regional

0.043862400217058756

http://rd.yahoo.com/search/navbar/top/news/manufacturer+automobile/*http://search.news.yahoo.com/search/news?p=manufacturer+automobile&hc=33&hs=1

2.980499412780923E-25

<http://search.yahoo.com/search/options?hc=33&hs=1620&p=manufacturer+automobile>

1.3703445576004248E-26

Level Four

http://rd.yahoo.com/search/navbar/top/goo/manufacturer+automobile/*http://search.yahoo.com/search?p=manufacturer+automobile&h=w&hc=33&hs=1620

0.2710842870269371

http://rd.yahoo.com/M=226514.2050602.3520739.1990504/D=yahoo/S=2766679:SE3/A=1072491/R=0/*http://tm.ask.com/r?t=c&s=a&id=30280&sv=&uid=&sid=&p=&o=0&u=http://www.ask.com/main/askjeeves.asp?ask=manufacturer+automobile

3.5995528921750565E-4

<http://search.yahoo.com/search/options?p=manufacturer+automobile&h=c>

5.056013708334478E-6

http://rd.yahoo.com/O=1/I=search/dyn/yp/manufacturer+automobile/*http://yp.yahoo.com/py/ypResults.py?stx=manufacturer+automobile

4.090401013835291E-6

Level Five

http://rd.yahoo.com/search/navbar/nextc/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&hc=33&hs=1&b=4&h=C&xargs=

0.12685697437960447

http://srd.yahoo.com/S=2766679:WS0/R=2/K=manufacturer+automobile/CS=7678473/*http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/Automotive/Motorcycles/Makers/

2.85821620801396E-19

http://srd.yahoo.com/S=2766679:WS0/R=1/K=manufacturer+automobile/CS=7853797/*http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/Automotive/Makers/

3.358654343182124E-20

<http://search.yahoo.com/search/options?p=manufacturer+automobile&h=w&hc=33&hs=1620>

2.9076467196631425E-20

Level Six

http://rd.yahoo.com/search/navbar/top/news/manufacturer+automobile/*http://search.news.yahoo.com/search/news?p=manufacturer+automobile&hc=33&hs=1620

0.17419823779264362

http://rd.yahoo.com/search/navbar/prevs/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&xargs=

0.37319636119269706

http://rd.yahoo.com/search/navbar/first/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&h=C

0.054324461775269235

<http://search.yahoo.com/search/options?p=manufacturer+automobile&hc=33&hs=1&b=4&h=C&xargs=&h=c>

4.3052446166305105E-4

Level Seven

http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search/dir?hc=33&hs=1620&p=manufacturer+automobile

0.31492186647230325

http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search?hc=33&hs=1620&p=manufacturer+automobile

0.20961902069480615

http://story.news.yahoo.com/news?tmpl=story&u=/ap/20021119/ap_on_go_pr_wh/child_passenger_safety_1

2.56407758366153E-27

http://biz.yahoo.com/opt/021119/eef7bfc8222e21259301721f97c53f5c_1.html

1.282038791830765E-27

http://srd.yahoo.com/S=2766679:DS0/R=1/TR=33/K=manufacturer+automobile/CS=7853797/*http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/Automotive/Makers/

4.622203125190925E-39

Level Eight

http://rd.yahoo.com/search/navbar/top/summary/manufacturer+automobile/*http://search.yahoo.com/search/dir?p=manufacturer+automobile

0.6153460024246546

http://rd.yahoo.com/search/navbar/regional/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&o=&g=&n=20&h=C&r=Regional

0.04140713812331551

http://rd.yahoo.com/search/navbar/top/news/manufacturer+automobile/*http://search.news.yahoo.com/search/news?p=manufacturer+automobile&hc=33&hs=1

5.1076752291678925E-24

<http://search.yahoo.com/search/options?hc=33&hs=1620&p=manufacturer+automobile>

2.3484767519062934E-25

A-2 “Automobile Manufacturer” Case (HITS)

Level One

<http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1620>

0.4223452874385104

http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/

0.00965370328620864

http://dir.yahoo.com/Regional/Countries/Canada/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/

0.00826283875052485

<http://www.japanauto.com/>

0.00588277876957169

<http://www.aiam.org/>

0.00468378645210461

<http://yp.yahoo.com/py/ypResults.py?stx=manufacturer+automobile>

0.00239185455486392

Level Two

[.http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile](http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile)

0.21442 164009203613

[.http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile](http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile)

0.19381539445954882

<http://yp.search.yahoo.com/search/ypredirect?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile>
0.08975808754671673

http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645
0.008756886589923582

http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645

0.006567664942442686

http://search.news.yahoo.com/search/news/?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile+&c=news_photos

0.039810864163511435

Level Three

<http://search.yahoo.com/search/options?hc=33&hs=1618&p=manufacturer+automobile>

7.033237582143429E-5

http://rd.yahoo.com/search/navbar/top/summary/manufacturer+automobile/*http://search.yahoo.com/search/dir?p=manufacturer+automobile

0.003736351949007416

http://rd.yahoo.com/search/navbar/top/news/manufacturer+automobile/*http://search.news.yahoo.com/search/news?p=manufacturer+automobile&hc=33&hs=1

0.00193350926259912

http://rd.yahoo.com/search/navbar/nextc/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&hc=33&hs=1&b=4&h=C&xargs=

0.036355332056876216

http://srd.yahoo.com/S=2766679:WS0/R=1/K=manufacturer+automobile/CS=7853797/*http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/Automotive/Makers/

3.453511542039411E-5

http://srd.yahoo.com/S=2766679:WS0/R=2/K=manufacturer+automobile/CS=7678473/*http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/Automotive/Motorcycles/Makers/

2.9417955356367297E-4

Level Four

<http://search.yahoo.com/search/options?p=manufacturer+automobile&h=c2.567160303678601E-5>

http://rd.yahoo.com/search/navbar/top/goo/manufacturer+automobile/*http://search.yahoo.com/search?p=manufacturer+automobile&h=w&hc=33&hs=1618

0.028409323469575868

http://rd.yahoo.com/search/navbar/top/news/manufacturer+automobile/*http://search.news.yahoo.com/search/news?p=manufacturer+automobile&hc=33&hs=1618

0.09866212423764942

http://rd.yahoo.com/search/navbar/next/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&hc=33&hs=1618&b=4&h=C&xargs=

0.3153239458221882

http://srd.yahoo.com/S=2766679:DS0/R=1/TR=33/K=manufacturer+automobile/CS=7853797/*http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/Automotive/Makers/

5.887518915971463E-4

http://srd.yahoo.com/S=2766679:DS0/R=2/TR=33/K=manufacturer+automobile/CS=7678473/*http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/Automotive/Motorcycles/Makers/

0.00501755479913726

Level Five

<http://search.yahoo.com/search/options?p=manufacturer+automobile&h=w&hc=33&hs=16181.1534406741325163E-4>

http://srd.yahoo.com/S=2766679:WS0/R=3/K=manufacturer+automobile/CS=8516129/*http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/Automotive/Recreational_Vehicles/Makers/

6.385748079884405E-4

http://rd.yahoo.com/search/navbar/regional/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&o=&g=&n=20&h=C&r=Regional

0.0024297544821472905

http://rd.yahoo.com/partner/2766679/overture/ca_first/OV=0/5/1/manufacturer+automobile/*http://www3.overture.com/d/sr/?xargs=00u3hs9yoakFUOzyDCxgwGn6aO%2FYow84OD9nAL4IYwRjNOIPjQdvZk05pKpfDbTT9fQ6O3DOOfRTcHBLVXgZ8hybW2eXrzgKk4aAwJdR6CnPodvpeYxJjvll5wVnAxgErMFALVISOxclYt95ysHZVPq00ohKBajo%2BcVileytkhsuK1Wq3U9%2FMs8FcWX9tEKTjwkP87oL1A%2FAKYXFjyGQXLz0mnD7LsZi59g4zuSOT3VzhLYpyEilPVn8o7y%2B3%2F8nVx%2BFDJd3pCoiePvQW3g1jJ0%3D

1.452519790334235E-4

http://rd.yahoo.com/partner/2766679/overture/ca_first/OV=0/5/3/manufacturer+automobile/*http://www3.overture.com/d/sr/?xargs=00u3hs9yoakE0OTSdChRBTT1PWbSiwXVSoLibAvMWORNgRQLxMgL3qTVZk%2B%2FM3vXWjKfYGMNxn3rRFdaDBV41DnNr0S5dnrDldm82ThfoztWafFzm8Xxciaa1%2Fx2FcYNQCtpQaEjl%2BdbmrLgLRqErrupSVB7wW8ohQyYGIW4wcPLQUjJkm%2Bf9hScf7dFkqGkKbjtUdqhsSqhNmXJJw73gxesjxU2VvP8FYUwL5Bbglk8%3D

1.473957685671414E-4

http://rd.yahoo.com/search/navbar/nexts/*http://search.yahoo.com/search?p=manufacturer+automobile&b=21&hc=33&hs=1&xargs=

0.03738692317020067

Level Six

http://srd.yahoo.com/S=8516129:D1/CS=8516129/SS=13002468/*http://www.northern-lite.com

7.030857640493626E-39

http://srd.yahoo.com/S=8516129:D1/CS=8516129/SS=2004353/*http://www.nuwa.com/

7.030857640493626E-39

<http://search.yahoo.com/search/options?p=manufacturer+automobile&o=&g=&n=20&h=C&r=Regional&h=c>

1.9192366788055605E-5

http://rd.yahoo.com/search/navbar/top/goo/manufacturer+automobile/*http://search.yahoo.com/search?p=manufacturer+automobile&o=&g=&n=20&r=Regional&h=w&hc=447&hs=974

0.001969366701482784

http://rd.yahoo.com/search/navbar/top/news/manufacturer+automobile/*http://search.news.yahoo.com/search/news?p=manufacturer+automobile&hc=447&hs=974

5.275644923188644E-4

<http://search.yahoo.com/search?p=manufacturer+automobile>

0.14810182671108818

Level Seven

<http://search.yahoo.com/search/options?p=manufacturer+automobile&o=&g=&n=20&r=Regional&h=w&hc=447&hs=974>

7.224889180540314E-5

http://rd.yahoo.com/search/navbar/top/summary/manufacturer+automobile/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&o=&g=&n=20&r=Regional

0.009607146132303184

http://rd.yahoo.com/search/navbar/top/news/manufacturer+automobile/*http://search.news.yahoo.com/search/news?p=manufacturer+automobile&hc=447&hs=1

0.03678339824641549

http://rd.yahoo.com/search/navbar/nextc/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&o=&g=&r=Regional&hc=447&hs=1&b=5&h=C&xargs=

0.25083950208101174

http://srd.yahoo.com/S=2766679:WS0/R=2/K=manufacturer+automobile/CS=22550670/*http://dir.yahoo.com/Regional/Countries/United_Kingdom/Business_and_Economy/Shopping_and_Services/Automotive/Makers/

6.64494351439294E-4

http://rd.yahoo.com/partner/2766679/overture/ca_first/OV=0/5/1/manufacturer+automobile/*http://www3.overture.com/d/sr/?xargs=00u3hs9yoakFUOzyDCxgwGn4aOeZ%2FDtIsOz0h8HoYBSTjEHDQNvb403UIRvhNptOv7bOfxTNrW05KGIPVwsETGybZyly7P2STbmM4VivQT52D9HD7cJe2ypXlguVlfZ0aGovadUXTpFcP473KknkwzZb4D2GuvaoE0LLe9w2wb7m1MLSrIFpZup7md%2F4EgspfaDpO7GcflgsiO2MAZHXPKEcQunvnB3PXHTFj%2FcbnJECeLDNlnKZQiMojoy9%2F%2F%2Byp%2FAcRpKcoQnDexXKeAvp39bVj

0.00167435859622094

Level Eight

<http://search.yahoo.com/search/options?p=manufacturer+automobile&o=&g=&n=20&r=Regional&h=c>

1.7385650969263598E-4

http://rd.yahoo.com/search/navbar/nextc/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&o=&g=&n=20&r=Regional&hc=447&hs=974&b=5&h=C&xargs=

0.004642613684422472

http://srd.yahoo.com/S=2766679:DS0/R=2/TR=447/K=manufacturer+automobile/CS=22550670/*http://dir.yahoo.com/Regional/Countries/United_Kingdom/Business_and_Economy/Shopping_and_Services/Automotive/Makers/

8.664142678842098E-5

http://rd.yahoo.com/partner/2766679/overture/ca_first/OV=0/5/1/manufacturer+automobile/*http://www3.overture.com/d/sr/?xargs=00u3hs9yoakFUOwSDChgwGn6aODoDYdhth7OEXwYhgXJocQfGg7e2JpppKp%2FDbTb%2BvMbNrxUjtjU07CGhPFkw7ljV1ksy7XnwxJikBwLbe6jnNufvaeUpxhuV15g7kDL7JTWa%2BEa64EZqTKn0whgb4Z9Wsm1QCW6xvecVhjO9K0yVsjK6mU6juZlvgrCy99ouk5sZB8uyC8JaJg3l4s5QziGY9o4Pdy%2FKYuf6PcbIgYNzCET1MehE4jV9%2FKu8tt%2Fw5VCfhRzeFOCoie3voy7QJjUK0%3D

2.1746405114474345E-4

http://rd.yahoo.com/partner/2766679/overture/ca_first/OV=0/5/3/manufacturer+automobile/*http://www3.overture.com/d/sr/?xargs=00u3hs9yoakE0OTSdChRBTtwPwoVgWC2Viwu1MC4xbUiaLFLeJsBedSWrMv35G98OeVpyWmMPgu3CCGu8eBXTlccHRSXrGS4uQL3a4rTWE5ZlzVF68HpCGeV18obCIUGNMkbVEAUUm10%2F5zp3mGoKKikAOXpq3k0qC3AwYHF6D7pthaMhvbIj2Vpx%2Ft0ySocIhadujtbrhNmExZ8sIAY2PJ6KPFzdWy%2FwRhTMvkyLrpk8%3D

2.1775087920264039E-4

http://rd.yahoo.com/search/navbar/nexts/*http://search.yahoo.com/search/dir?p=manufacturer+automobile&o=&g=&n=20&r=Regional&b=21&h=s&xargs=

0.15942314276104352

B-1 “Java Developer” Case (ES)

Level One

<http://www.ibm.com/developerworks/java/>

0.4128629605998993

<http://java.sun.com/>

0.03965328753334086

<http://developer.java.sun.com>

0.02427424567471161

<http://www.javalobby.org/>

0.01184420477709548

<http://www.jfind.com/javadev.shtml>

0.00771643412249827

<http://www.javaalmanac.com/>

0.00393804118843172

Level Two

<http://www-106.ibm.com/developerworks/theme/>

0.21389528634262828

<http://www-106.ibm.com/developerworks/java/library/l-secjav.html>

0.20174847712950947

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.01364113199828978

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.07649925899708516

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www7b.boulder.ibm.com/wsdd/&origin=j>

0.0226409369130074

<http://www.symbian.com/news/2002/expo03-feat.html>

0.00287951764014932

Level Three

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/?origin=dwheader>

0.001385470878252785

<http://www-106.ibm.com/developerWorks/ibm>

0.32168889328525235

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/websphere/developer&origin=devdomain>

0.01245535194411498

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7b.software.ibm.com/dmdd/&origin=devdomain>

0.009252106433399163

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7.software.ibm.com/vad.nsf/&origin=devdomain>

0.004893986447348429

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/ibm/?origin=dwheader>

7.280153202756514E-4

0.02675688112399534

Level Four

<http://www-106.ibm.com/developerworks/ibm/library/i-jmsmq/>

0.1749577103508998

<http://www-106.ibm.com/developerworks/ibm/library/i-asyncmsg/>

0.2269783093473772

<http://www-106.ibm.com/developerworks/ibm/library/i-jms2/>

0.17142661663547695

<http://www-106.ibm.com/developerworks/ibm/library/i-2extreme1/>

0.1118853052609475

<http://www-106.ibm.com/developerworks/newsletter/>

0.04976426968298338

<http://www-106.ibm.com/developerworks/ibm/library/i-jbucks/>

0.0018754403883559244

Level Five

<http://www-105.ibm.com/developerworks/papers.nsf/ibm-papers-bytitle?OpenView&Count=500>

0.35569163696738243

<http://www-106.ibm.com/developerworks/ibm/>

0.0923671662382465

<http://www.ibm.com/developerworks/ibm/>

0.0923671662382465

<http://www-106.ibm.com/developerworks/>

0.024957051492033092

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.0014966621260334766

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www-3.ibm.com/software/ts/mqseries/txppacs/ma88.html&origin=devdomain>

0.0011240811068330679

Level Six

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.005254136849335422

<http://www-106.ibm.com/developerworks/java/>

0.6410631063873666

<http://www-105.ibm.com/developerworks/education.nsf/xml-onlinecourse-bytitle/DC1BA2350F8A7BAF86256C6A0046BD72?OpenDocument>

0.06714950781407919

<http://www-106.ibm.com/developerworks/java/library/j-mocktest.html>

0.040004640433319236

<http://www-105.ibm.com/developerworks/education.nsf/xml-onlinecourse-bytitle/25027A9A440EF26186256C7800508614?OpenDocument>

0.0114560819908198

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/?origin=dwheader>

7.295866049506737E-5

Level Seven

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/ibm/?origin=dwheader>

4.744252957849326E-4

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/privacy/?origin=dwheader>
4.744252957849326E-4

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/legal/?origin=dwheader>
4.744252957849326E-4

<http://www-106.ibm.com/developerworks/country/>
0.36566496321403763

<http://www-106.ibm.com/developerworks/xml/>
0.028132541121973734

<http://www-106.ibm.com/developerworks/java/library/j-xpcol.html>
8.398117482765826E-4

Level Eight

<http://www-106.ibm.com/developerworks/newsletter/>
0.28621169549107944

<http://www.ibm.com/Privacy/>
7.192271767444468E-4

<http://www-106.ibm.com/developerworks/feedback/>
0.004817070630790561

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>
0.0016374196955629035

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/&origin=dwheader>
4.562254742887719E-5

<http://www.ibm.com/investor/financials/irfiar.phtml>
2.3724792205929718E-74

B-2 “Java Developer” Case (HITS)

Level One

<http://www.ibm.com/developerworks/java/>
0.4128629605998993

<http://java.sun.com/>
0.03965328753334086

<http://developer.java.sun.com>
0.02427424567471161

<http://www.javalobby.org/>
0.01184420477709548

<http://www.jfind.com/javadev.shtml>

0.00771643412249827

<http://www.javaalmanac.com/>

0.00393804118843172

Level Two

<http://www-106.ibm.com/developerworks/java/library/j-jsp04153.html>

0.12385042367469752

<http://www-106.ibm.com/developerworks/java/library/j-jstl0415/>

0.0816320479604754

<http://www-106.ibm.com/developerworks/java/library/j-newviews/>

0.048788759036143776

<http://www-106.ibm.com/developerworks/java/library/j-jtp04223.html>

0.04160064859367833

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.039942197195027083

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.0028765258598279033

Level Three

http://www-132.ibm.com/content/home/store_IBMPublicUSA/en_US/navcode_840.html

3.257179000476189E-107

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7.software.ibm.com/vad.nsf/&origin=devdomain>

9.249418971005956E-6

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7b.software.ibm.com/dmdd/&origin=devdomain>

1.604317068638799E-5

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/?origin=dwheader>

3.470121200696845E-5

<http://www-106.ibm.com/developerWorks/ibm>

0.0021562797630789465

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/websphere/developer&origin=devdomain>

2.3293613181529937E-5

Level Four

<http://www.ibm.com/contact/>

6.657562515391704E-26

<http://www.ibm.com/contact/employees/>

6.329615126076059E-53

<http://www.ibm.com/>

2.3133913357291708E-32

http://www-919.ibm.com/developer/db2/documents/data_acc/index.html

5.425392800335183E-24

<http://www-106.ibm.com/developerworks/newsletter/?loc=db2dd>

0.0019992703893809516

<http://www-106.ibm.com/developerworks/ibm/library/i-ptoolkit.html/>

0.16316307068340888

Level Five

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.0010521986663254448

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.01830929851781638

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/ibm/?origin=dwheader>

7.17441458638789E-4

<http://www-106.ibm.com/developerworks/feedback/>

0.01740639417937311

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www-3.ibm.com/software/mailing-lists/newsletter/>

1.0838057958781989E-4

http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www.developer.ibm.com/news/newsletter_ann.html

0.0013400601882804066

Level Six

http://www-132.ibm.com/content/home/store_IBMPublicUSA/en_US/navcode_840.html

1.2868094241165783E-46

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7.software.ibm.com/vad.nsf/&origin=devdomain>

0.006793227352868294

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7b.software.ibm.com/dmdd/&origin=devdomain>

0.01176878962025849

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/?origin=dwheader>

0.0010579664894969293

<http://www-106.ibm.com/developerWorks/ibm>

0.11260318775662524

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/websphere/developer&origin=devdomain>

0.017120399852189907

Level Seven

<http://www.ibm.com/contact/>

6.657562515391704E-26

<http://www.ibm.com/contact/employees/>

6.329615126076059E-53

<http://www.ibm.com/>

2.3133913357291708E-32

http://www-919.ibm.com/developer/db2/documents/data_acc/index.html

5.425392800335183E-24

<http://www-106.ibm.com/developerworks/newsletter/?loc=db2dd>

0.0019992703893809516

<http://www-106.ibm.com/developerworks/ibm/library/i-ptoolkit.html/>

0.16316307068340888

Level Eight

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>
0.0010521986663254448

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.01830929851781638

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/ibm/?origin=dwheader>

7.17441458638789E-4

<http://www-106.ibm.com/developerworks/feedback/>

0.01740639417937311

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www-3.ibm.com/software/mailing-lists/newsletter/>

1.0838057958781989E-4

http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www.developer.ibm.com/news/newsletter_ann.html

0.0013400601882804066

C-1 “Artificial Intelligence” Case (ES)

Level One

http://dir.yahoo.com/Entertainment/Movies_and_Film/Titles/Science_Fiction_and_Fantasy/Artificial_Life_Forms/A_I__Artificial_Intelligence/

0.1886505673185579

http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/

0.18073010550157712

http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/European_Coordinating_Committee_for_Artificial_Intelligence_ECCAI/

0.07655744993755742

<http://www.ai.mit.edu/>

0.002408330522024467

<http://www1.overture.com/d/sr/?xargs=02u3hs9yoaj00SvTCChBDFXx0qBc45AxSKk%2FLC2ogXaCEwZghg%2BMxq9gZg2q93YtvvL3nCm83geDEa1UU%2Fpsyf4BZGiVRMJjybYzDF2HxQAZBbqJQ5QUWQOQBfKLGCLeLDHo8w78Z1jNNiAsml1pWTJcFJCvAjUc0DkRmkCtEuDf4I9j2as76q3WD9MG9vacb3E3rKGJUFFqxDOpqlP9Dh23%2BPZwbk%2B1ICNknmy1Oc7%2BBaABV>

0.00124 25659234176027

<http://www.aaai.org/>

0.0010729828466247205

Level Two

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=42779322/*http://www.rottentomatoes.com/movie-10000163/about.php

0.3581355972297218

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=27550374/*http://www.dreamworksfansite.com/ai/

0.00632 6540459276429

http://srd.yahoo.com/S=245960:D1/CS=245960/SS=42656828/*http://www.singinst.org/

0.006093600122767582

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=58521013/*http://www.filmtracks.com/titles/ai.html

0.003017693639517619

http://srd.yahoo.com/S=245960:D1/CS=245960/SS=245962/*http://www.aaai.org/

0.0018874133144593935

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=39249876/*http://aimovie.warnerbros.com/

0.0012 16582316065081

Level Three

http://rd.yahoo.com/search/navbar/top/summary/intelligence+artificial/*http://search.yahoo.com/search/dir?p=intelligence+artificial

0.6195152025111741

http://rd.yahoo.com/search/navbar/regional/*http://search.yahoo.com/search/dir?p=intelligence+artificial&o=&g=&n=20&h=C&r=Regional

0.039649235357224026

http://rd.yahoo.com/search/navbar/top/news/intelligence+artificial/*http://search.news.yahoo.com/search/news?p=intelligence+artificial&hc=6&hs=1

5.962046678869396E-37

<http://search.yahoo.com/search/options?hc=6&hs=258&p=intelligence+artificial>

2.2288024967736065E-38

Level Four

http://rd.yahoo.com/O=1/l=search/dyn/news/intelligence+artificial/*http://search.news.yahoo.com/search/news?n=20&c=news&p=intelligence+artificial

0.1441780392356

http://rd.yahoo.com/O=1/l=search/dyn/shopping/intelligence+artificial/*http://search.shopping.yahoo.com/search?P=all&is=1&p=intelligence+artificial

0.13757992296719945

http://rd.yahoo.com/M=226514.2050602.3520739.1990504/D=yahoo/S=2766679:SE3/A=1072491/R=0/*http://tm.ask.com/r?t=c&s=a&id=30280&sv=&uid=&sid=&p=&o=0&u=http://www.ask.com/main/askjeeves.asp?ask=intelligence+artificial

0.07041434658781436

http://rd.yahoo.com/O=1/l=search/dyn/auctions/intelligence+artificial/*http://search.auctions.yahoo.com/search/auc?alocale=0us&acc=us&p=intelligence+artificial

0.016676104019178836

http://rd.yahoo.com/O=1/l=search/dyn/movies/intelligence+artificial/*http://search.movies.yahoo.com/search/moviesintgsearch?p=intelligence+artificial

0.0018529004465754264

Level Five

http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search?n=20&p=intelligence+artificial

0.3219846645871931

http://rd.yahoo.com/search/navbar/top/*http://search.yahoo.com/search/dir?n=20&p=intelligence+artificial

0.2984824562461546

http://story.news.yahoo.com/news?tmpl=story&u=/nm/20021114/ts_nm/attack_congress_dc_1

9.114226171915044E-19

http://story.news.yahoo.com/news?tmpl=story&u=/dowjones/20021120/bs_dowjones/200211200040000039

2.604064620547155E-19

Level Six

http://rd.yahoo.com/search/navbar/top/summary/intelligence+artificial/*http://search.yahoo.com/search/dir?n=20&p=intelligence+artificial

0.2625099562508695

http://rd.yahoo.com/search/navbar/nextc/*http://search.yahoo.com/search/dir?p=intelligence+artificial&hc=6&hs=1&b=4&h=C&xargs=00u3hs9yoaKCstSjWzNVAAmrvjzLJy1NcwzBQABfZ%2FgAc%3D

0.16824071569902138

http://rd.yahoo.com/search/navbar/top/news/intelligence+artificial/*http://search.news.yahoo.com/search/news?p=intelligence+artificial&hc=6&hs=1

8.81690700845398E-26

<http://search.yahoo.com/search/options?n=20&p=intelligence+artificial>

3.296040003160367E-27

Level Seven

http://rd.yahoo.com/search/navbar/top/goo/intelligence+artificial/*http://search.yahoo.com/search?n=20&p=intelligence+artificial&h=w&hc=6&hs=258

0.18691772402312812

http://rd.yahoo.com/search/navbar/next/*http://search.yahoo.com/search/dir?n=20&p=intelligence+artificial&hc=6&hs=258&b=4&h=C&xargs=

0.04848318732185068

http://rd.yahoo.com/search/navbar/top/news/intelligence+artificial/*http://search.news.yahoo.com/search/news?p=intelligence+artificial&hc=6&hs=258
6.984280226756426E-25

<http://search.yahoo.com/search/options?n=20&p=intelligence+artificial&h=c>

9.427710939761145E-27

Level Eight

http://srd.yahoo.com/S=2766679:WS0/R=1/K=intelligence+artificial/CS=245765/*http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/

0.19895689759914653

http://srd.yahoo.com/S=2766679:WS0/R=2/K=intelligence+artificial/CS=519794/*http://dir.yahoo.com/Recreation/Games/Computer_Games/Internet_Games/Web_Games/Artificial_Intelligence/

0.0797672565706376

http://srd.yahoo.com/S=2766679:WS0/R=3/K=intelligence+artificial/CS=21390/*http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Computers/Software/Artificial_Intelligence/

5.116849294694656E-13

<http://search.yahoo.com/search/options?n=20&p=intelligence+artificial&h=w&hc=6&hs=258>

3.860717468150723E-14

C-2 “Artificial Intelligence” Case (HITS)

Level One

http://dir.yahoo.com/Entertainment/Movies_and_Film/Titles/Science_Fiction_and_Fantasy/Artificial_Life_Forms/A_I__Artificial_Intelligence/

0.1886505673185579

http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/

0.18073010550157712

http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/European_Coordinating_Committee_for_Artificial_Intelligence__ECCAI/

0.07655744993755742

<http://www.ai.mit.edu/>

0.002408330522024467

<http://www1.overture.com/d/sr/?xargs=02u3hs9yoaj00SvTCChBDFXx0qBc45AxSKk%2FLC2ogXaCEwZghg%2BMxq9gZg2q93Yt vvL3n Cm83geDEa1UU%2Fptsyf4BZGiVRMJjybYzDF2HxQAZBbqJQ5QUWQOQBfKLGcLeLDHo8w78ZijNNiAsmlIpWTJcFIC vAjUc0DkRmkCtEuDf419j2as76q3WD9MG9vacb3E3rKGJUFFqxDOpquiP9Dh23%2BPZwBk%2B1ICNknmylOc7%2BBaABV>

0.00124 25659234176027

<http://www.aaai.org/>

0.0010729828466247205

Level Two

<http://www.aaai.org/AITopics/html/current.html>

0.3727713512640388

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=42779322/*http://www.rottentomatoes.com/movie-10000163/about.php

0.0622171040374019

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=27550374/*http://www.dreamworksfansite.com/ai/

0.0061134539539702375

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=58521013/*http://www.filmtracks.com/titles/ai.html

0.004075635969313492

http://srd.yahoo.com/S=245960:D1/CS=245960/SS=245962/*http://www.aaai.org/

0.0037054488626334244

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=39249876/*http://aimovie.warnerbros.com/

0.002037817984656746

Level Three

<http://mtf.news.yahoo.com/mailto?url=http://movies.yahoo.com/shop%3fd=hv%26cf=info%26id=1804383571&title=A.I.%3a+Artificial+Intelligence&prop=movieinfo&locale=us&hl=info>

1.0278431368719676E-38

<http://www.qksrv.net/click-637882-5463217?loc=http%3A//search.ebay.com/search/search.dll%3FmficISAPICCommand%3DGetResult%26ht%3D1%26SortProperty%3DMetaEndSort%26query%3Dartificial+intelligence>

0.3559052726084767

http://www.amazon.com/exec/obidos/ASIN/0312280610/o/qid=990202631/sr=8-1/ref=aps_sr_b_1_1/stevenspielbefan/107-3769883-5568548

7.0868962190741955E-31

<http://www.filmtracks.com/ratings/>

8.597307820382831E-43

<http://www.aimovie.com/>

9.831343463827147E-28

<http://search.yahoo.com/search/options?hc=6&hs=258&p=intelligence+artificial>

1.9922205316903735E-4

Level Four

<http://movies.yahoo.com/shop?d=hv&cf=info&id=1804383571>

2.1822576349372274E-70

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1>

3.460653118538246E-4

<http://search-completed.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2&SortProperty=MetaEndSort>
0.0163887676197924

<http://cgi1.ebay.com/aw-cgi/eBayISAPI.dll?AddSavedSearch&savedsearch=http://search.ebay.com/search/search.dll?GetResult%26st%3d2%26SortProperty%3dMetaEndSort%26query%3Dartificial%2Bintelligence%26ht%3D1%26combine%3Dy>

7.574175230201142E-4

<http://search.stores.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&rd=on&fp=4>
0.01114063193256214

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1580777706>

3.460653118538246E-4

Level Five

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1&rflsActive=0>

2.699515137852154E-4

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2>

0.027000812614401293

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1572728430>

0.014192556818996454

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1572731162>

0.017031068182795744

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1572734779>

0.017031068182795744

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1572741260>

0.0056770227275985815

Level Six

<http://cgi1.ebay.com/aw-cgi/eBayISAPI.dll?AddSavedSearch&savedsearch=http://search.ebay.com/search/search.dll?GetResult%26st%3d2%26query%3Dartificial%2Bintelligence%26ht%3D1%26combine%3Dy>

0.0015905466935944872

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=920239832>

0.0035615384681995437

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=920239841>

0.00581320634684869

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1579909277>

0.004469608409091587

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1581712834>

0.0022275545454873203

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1579961446>

0.004737981880401997

Level Seven

<http://www.andale.com/gallery?cid=8066829&mid=1>

9.755258480356855E-12

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1>

1.4410803379369142E-80

<http://search-completed.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2&SortProperty=MetaEndSort>

0.2560709046264007

<http://search.stores.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&rd=on&fp=4>

0.16408460773931471

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1581765742>

0.027606373909953557

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1580777706>

2.4634336313591067E-12

Level Eight

<http://cgi.ebay.com/aw-cgi/eBayISAPI.dll?ViewItem&item=1579909277>

2.9242110294811343E-33

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1&rflsActive=0>

2.744501402122463E-4

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2>

0.027450769599177076

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1572728430>

0.014429069703394923

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1572731162>

0.01731488364407391

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=1572734779>

0.01731488364407391

D-1 “WWW conference” Case (ES)

Level One

<http://search.news.yahoo.com/search/news?p=conference+www+&c=news>

0.30751 61142110977

<http://www.sys-con.com/webwest/>

0.11375 166226648216

<http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=31>

0.089570527213745507

<http://www.iisg.nl/~w3vl/>

0.01277 8443675237054

<http://www.www2002.org/>

0.00996 7715504353978

<http://www oulu.fi/~spaceweb/lib/>

0.00704 1720594491317

Level Two

<http://search.yahoo.com/search/dir?c=news&ei=UTF-8&p=conference+www>

0.44631382108011247

<http://yp.search.yahoo.com/search/ypredirect?c=news&ei=UTF-8&p=conference+www>

0.08895520277167124

http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dconference%2Bwww%2B%26c%3Dnews

0.010107621002417598

http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dconference%2Bwww%2B%26c%3Dnews

0.006765269972817512

http://rd.yahoo.com/M=224039.1983422.3465437.2905688/D=news/P=m2n1llcf13000500/S=53720272:HEAD/A=1030394/R=0/*http://news.yahoo.com

0.005436305193304572

http://rd.yahoo.com/M=243098.2885920.4233620.2839171/D=news/S=53720272:PB/A=1354070/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=243098.2885920.4233620.2839171/D=news/S=53720272:PB/A=1354070/R=5/1051061649+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/poweredby/smb/index-yahoo.html

0.0010872610386609145

Level Three

http://srd.yahoo.com/S=2766679:WS3/R=1/K=conference+www/*http://search.yahoo.com/search?p=conference+www&hq=site:www.ncsl.org

0.35741427304586987

http://rd.yahoo.com/search/navbar/top/summary/conference+www/*http://search.yahoo.com/search/dir?p=conference+www

0.1942145878725122

http://rd.yahoo.com/search/navbar/top/news/conference+www/*http://search.news.yahoo.com/search/news?p=conference+www&hc=1&hs=1

7.192313465805947E-7

http://srd.yahoo.com/S=2766679:WS0/R=1/K=conference+www/CS=77208/*http://dir.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Conferences/

2.644066635185144E-7

http://srd.yahoo.com/S=2766679:WS2/R=1/K=conference+www/CS=139786/*http://dir.yahoo.com/Government/U_S__Government/State_Government/Organizations/

2.286947625173427E-7

<http://search.yahoo.com/search/options?p=conference+www>

4.26651107514925E-8

Level Four

http://rd.yahoo.com/search/navbar/top/goo/conference+www/*http://search.yahoo.com/search?p=conference+www&h=w&hc=1&hs=27

0.1462539970434026

http://rd.yahoo.com/search/navbar/top/news/conference+www/*http://search.news.yahoo.com/search/news?p=conference+www&hc=1&hs=27

0.05375425865942327

http://rd.yahoo.com/M=224039.2393051.3817444.2329190/D=yahoo_top/S=2716161:HEAD/A=1241381/R=0/*http://www.yahoo.com

0.0018420163737709228

http://rd.yahoo.com/M=237312.2560641.3949934.569744/D=yahoo/S=2766679:N/A=1282723/*http://www.classmates.com/index.tfs=7239

4.776876122205777E-4

<http://search.yahoo.com/search/options?p=conference+www&h=c>

4.2440330837310267E-4

http://rd.yahoo.com/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=5/1037831065+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-uri/http://www.compaq.com/bridge/yahoo_poweredby/

7.675068224045495E-5

Level Five

http://rd.yahoo.com/M=224039.1984929.3466713.1922510/D=yahoo/P=f2f06akd11000900/S=2766679:HEAD/A=1030697/R=0/*http://www.yahoo.com

0.014838840544799423

http://rd.yahoo.com/search/navbar/nests/*http://search.yahoo.com/search?p=conference+www&b=21&hc=1&hs=1&xargs=0

0.4292790609211374

http://srd.yahoo.com/S=2766679:WS1/R=2/K=conference+www/SS=2243065/OCS=253475/T=1037831084/F=98c5d375e450945a65eb94ddd25379c/*http://www.conference-board.org/

0.005813006178362968

<http://search.yahoo.com/search/options?p=conference+www&h=w&hc=1&hs=27>

0.0013926313345262416

http://srd.yahoo.com/S=2766679:WS1/R=1/K=conference+www/SS=139794/OCS=135823/T=1037831084/F=e7b0a0de0bdd169766128387d2f854e1/*http://www.ncsl.org/

0.0013926313345262416

http://rd.yahoo.com/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=5/1037831084+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/2.5635575170681213E-4

Level Six

http://rd.yahoo.com/search/navbar/first/*http://search.yahoo.com/search?p=conference+www&hc=1&hs=1&xargs=

0.4865781665459795

http://rd.yahoo.com/search/navbar/top/summary/conference+www/*http://search.yahoo.com/search/dir?p=conference+www&xargs=0

0.17354411290262553

<http://search.yahoo.com/search/options?p=conference+www&b=21&hc=1&hs=1&xargs=0>

2.3955878993723907E-5

http://rd.yahoo.com/search/navbar/top/news/conference+www/*http://search.news.yahoo.com/search/news?p=conference+www&hc=1&hs=1

1.2651041934602643E-20

http://rd.yahoo.com/M=238520.2536902.4010786.572210/D=yahoo/S=2766679:N/A=1128634/*http://www.overture.com/d/advertisers/p/bjump/?o=U3871&b=5

6.97474906404486E-21

http://rd.yahoo.com/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=5/1037831095+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/

1.384203364191255E-22

Level Seven

http://rd.yahoo.com/M=237312.2560641.3949934.569744/D=yahoo/S=2766679:N/A=1282723/*http://www.classmates.com/index.tfs=7239

8.892209714435795E-4

http://rd.yahoo.com/search/navbar/top/goo/conference+www/*http://search.yahoo.com/search?p=conference+www&h=w&hc=1&hs=27

0.29204974185771937

http://rd.yahoo.com/search/navbar/top/news/conference+www/*http://search.news.yahoo.com/search/news?p=conference+www&hc=1&hs=27

0.013780170815045024

http://rd.yahoo.com/M=224039.1984929.3466713.1922510/D=yahoo/S=2766679:HEAD/A=1030697/R=0/*http://www.yahoo.com

0.011800301721021639

<http://search.yahoo.com/search/options?p=conference+www&xargs=0&h=c>

8.014503417720771E-4

http://rd.yahoo.com/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=5/1037831115+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/

1.452517315318377E-4

Level Eight

http://rd.yahoo.com/search/navbar/nxts/*http://search.yahoo.com/search?p=conference+www&b=21&hc=1&hs=1&xargs=0

0.2184717239199072

http://rd.yahoo.com/search/navbar/top/summary/conference+www/*http://search.yahoo.com/search/dir?p=conference+www

0.09041674809609598

<http://search.yahoo.com/search/options?p=conference+www&h=w&hc=1&hs=27>

3.3013028413203852E-6

http://srd.yahoo.com/S=2766679:WS0/R=1/K=conference+www/CS=77208/*http://dir.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Conferences/

2.3358415520593163E-25

http://srd.yahoo.com/S=2766679:WS1/R=1/K=conference+www/SS=139794/OCS=135823/T=1037831132/F=459c3b7c9913accabbbdfbeef8705404/*http://www.ncsl.org/

3.7703050716419885E-26

http://rd.yahoo.com/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=5/1037831132+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/

1.738460758394217E-27

D-2 “WWW Conference” Case (HITS)

Level One

<http://search.news.yahoo.com/search/news?p=conference+www+&c=news>

0.3075161142110977

<http://www.sys-con.com/webwest/>

0.11375 166226648216

<http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=31>

0.089570527213745507

<http://www.iisg.nl/~w3vl/>

0.01277 8443675237054

<http://www.www2002.org/>

0.00996 7715504353978

<http://www oulu.fi/~spaceweb/lib/>

0.00704 1720594491317

Level Two

<http://www2002.org>

0.048551716263302314

<http://www.w3.org>

0.010579748743324756

<http://www.apple.com>

0.056204915198912761

<http://www.research.ibm.com>

0.023143200376022903

<http://www.htdc.org>

0.001322468592915594

<http://www.verity.com>

0.001674890334598886

Level Three

http://rd.yahoo.com/M=224039.1984929.3466713.2148332/D=yahoo/S=2766679:HEAD/A=1030697/R=0/*http://www.yahoo.com

2.65067178376182E-4

http://rd.yahoo.com/M=237312.2560641.3949934.502794/D=yahoo/S=2766679:N/A=1282723/*http://www.classmates.com/index.tfs=7239

5.765531208473857E-5

<http://search.yahoo.com/search/options?hc=1&hs=26&p=conferences+www>

1.3717519230747362E-4

http://rd.yahoo.com/M=179842.1351601.3214785.2276645/D=yahoo/S=2766679:PB/A=1168887/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?compaq2+shopping:dmad/M=179842.1351601.3214785.2276645/D=yahoo/S=2766679:PB/A=1168887/R=5/1036770380+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/

7.462127896227599E-6

http://rd.yahoo.com/search/navbar/top/summary/conferences+www/*http://search.yahoo.com/search/dir?p=conferences+www

0.00731423321275709

http://rd.yahoo.com/search/navbar/top/news/conferences+www/*http://search.news.yahoo.com/search/news?p=conferences+www&hc=1&hs=1

5.284636298177856E-4

Level Four

http://t.yahoo.com/gc/yahootrk=p:s,a:r,s:2716149,l:mktpl/*http://rd.yahoo.com/M=233467.2381293.3996454.2316829/D=yahoo_top/S=2716149:M3/A=1311104/R=1/O=1/l=yahoo-fp-marketplace/overstock-watches/*http://shop.store.yahoo.com/cgi-bin/clink?overstock3+shopping:dmfp.marketplace/overstock-watches+http://us.rmi.yahoo.com/rmi/http://www.overstock.com/rmi-framed-url/http://www.overstock.com/cgi-bin/d2.cgi%3FPAGE=STATICPAGE%26PAGE_ID=406%26cid=24184%26fp=f

1.5280233819060446E-15

http://rd.yahoo.com/M=179842.1351578.2962872.1288581/D=yahoo_top/S=2716149:PB/A=1246724/R=0/*http://shop.store.yahoo.com/cgi-bin/clink?compaq2+shopping:dmd/M=179842.1351578.2962872.1288581/D=yahoo_top/S=2716149:PB/A=1246724/R=1/1036770422+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/

1.7089704685276873E-14

http://t.yahoo.com/gc/yahootrk=p:s,a:r,s:2716149,l:mktpl/*http://rd.yahoo.com/M=224464.2339675.3996837.2418683/D=yahoo_top/S=2716149:M2/A=1311126/R=0/O=1/l=yahoo-fp-marketplace/smartbargains-rldown/*http://shop.store.yahoo.com/cgi-bin/clink?smartbargains2+shopping:dmfp.marketplace/smartbargains-rldown+http://us.rmi.yahoo.com/rmi/http://www.smartbargains.com/rmi-framed-url/http://service.bfast.com/bfast/click%3Fbfbmid=35212156%26siteid=40069129%26bfp=y27

1.7089704685276873E-14

http://t.yahoo.com/gc/yahootrk=p:s,a:r,s:2716149,l:mktpl/*http://rd.yahoo.com/M=224464.2339675.3996837.2418683/D=yahoo_top/S=2716149:M2/A=1311126/R=1/O=1/l=yahoo-fp-marketplace/smartbargains-rldown/*http://shop.store.yahoo.com/cgi-bin/clink?smartbargains2+shopping:dmfp.marketplace/smartbargains-rldown+http://us.rmi.yahoo.com/rmi/http://www.smartbargains.com/rmi-framed-url/http://service.bfast.com/bfast/click%3Fbfbmid=35212156%26siteid=40069129%26bfp=y27

1.7089704685276873E-14

http://t.yahoo.com/gc/yahootrk=p:s,a:r,s:2716149,l:mktpl/*http://rd.yahoo.com/M=233467.2381293.3996454.2316829/D=yahoo_top/S=2716149:M3/A=1311104/R=0/O=1/l=yahoo-fp-marketplace/overstock-watches/*http://shop.store.yahoo.com/cgi-bin/clink?overstock3+shopping:dmfp.marketplace/overstock-watches+http://us.rmi.yahoo.com/rmi/http://www.overstock.com/rmi-framed-url/http://www.overstock.com/cgi-bin/d2.cgi%3FPAGE=STATICPAGE%26PAGE_ID=406%26cid=24184%26fp=f

1.7089704685276873E-14

<http://www.classmates.com/index.tf?s=4038>

6.412455104008887E-29

Level Five

<http://www.yahoo.com>

2.516435735327837E-8

<http://www.classmates.com/p.tf/registration/?s=4038>

4.7597230338196576E-35

http://rd.yahoo.com/M=239154.2595801.3977717.2247483/D=yahoo/S=2766679:N/A=1201210/R=2/id=noscript/*http://store.yahoo.com

7.336498244324962E-47

<http://search.yahoo.com/search/options?p=conferences+www&h=c>

9.448280374972562E-63

http://rd.yahoo.com/M=179842.1351601.3214785.2276645/D=yahoo/S=2766679:PB/A=1168887/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?compaq2+shopping:dmad/M=179842.1351601.3214785.2276645/D=yahoo/S=2766679:PB/A=1168887/R=5/1036770425+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/

2.4658752392273877E-9

http://rd.yahoo.com/search/navbar/top/goo/conferences+www/*http://search.yahoo.com/search?p=conferences+www&h=w&hc=1&hs=26

0.4686269478687704

Level Six

http://rd.yahoo.com/M=179842.1351578.2962872.1288581/D=yahoo_top/S=2716149:PB/A=1246724/R=0/*http://shop.store.yahoo.com/cgi-bin/clink?compaq2+shopping:dmad/M=179842.1351578.2962872.1288581/D=yahoo_top/S=2716149:PB/A=1246724/R=1/1036770476+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/

1.5938488591813986E-30

http://rd.yahoo.com/M=224039.2285323.3709562.2223080/D=st/P=ml00a18b01000c00/S=42979443:HEAD/A=1208012/R=1/*http://www.yahoo.com

2.3880445890016914E-15

<http://rd.yahoo.com/store/nav/?http://www.strouds.com/>

2.6533828766685355E-16

<http://rd.yahoo.com/store/nav/?http://www.alight.com/>

2.6533828766685355E-16

<http://rd.yahoo.com/store/nav/?http://www.gojane.com/>

1.3266914383342677E-14

Level Seven

http://t.yahoo.com/gc/yahootrk=p:s,a,r,s:2716149,l:mktpl/*http://rd.yahoo.com/M=233467.2381293.3996454.2316829/D=yahoo_top/S=2716149:M3/A=1311104/R=1/O=1/I=yahoo-fp-marketplace/overstock-watches/*http://shop.store.yahoo.com/cgi-bin/clink?overstock3+shopping:dmfp.marketplace/overstock-watches+http://us.rmi.yahoo.com/rmi/http://www.overstock.com/rmi-framed-url/http://www.overstock.com/cgi-bin/d2.cgi%3FPAGE=STATICPAGE%26PAGE_ID=406%26cid=24184%26fp=f

3.4060944595066145E-4

http://rd.yahoo.com/M=179842.1351578.2962872.1288581/D=yahoo_top/S=2716149:PB/A=1246724/R=0/*http://shop.store.yahoo.com/cgi-bin/clink?compaq2+shopping:dmad/M=179842.1351578.2962872.1288581/D=yahoo_top/S=2716149:PB/A=1246724/R=1/1036770488+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/

0.002395487418086108

http://t.yahoo.com/gc/yahootrk=p:s,a,r,s:2716149,l:mktpl/*http://rd.yahoo.com/M=224464.2339675.3996837.2418683/D=yahoo_top/S=2716149:M2/A=1311126/R=0/O=1/I=yahoo-fp-marketplace/smartbargains-rldown/*http://shop.store.yahoo.com/cgi-bin/clink?smartbargains2+shopping:dmfp.marketplace/smartbargains-rldown+http://us.rmi.yahoo.com/rmi/http://www.smartbargains.com/rmi-framed-url/http://service.bfast.com/bfast/click%3Fbmid=35212156%26siteid=40069129%26bfp=y27

0.002395487418086108

http://t.yahoo.com/gc/yahootrk=p:s,a,r,s:2716149,l:mktpl/*http://rd.yahoo.com/M=224464.2339675.3996837.2418683/D=yahoo_top/S=2716149:M2/A=1311126/R=1/O=1/I=yahoo-fp-marketplace/smartbargains-rldown/*http://shop.store.yahoo.com/cgi-

bin/clink?smartbargains2+shopping:dmfp.marketplace/smartbargains-
rldown+http://us.rmi.yahoo.com/rmi/http://www.smartbargains.com/rmi-framed-
url/http://service.bfast.com/bfast/click%3Fbfbmid=35212156%26siteid=40069129%26bfp=ym27

0.002395487418086108

http://t.yahoo.com/gc/yahootrk=p:s,a:r,s:2716149,l:mktpl/*http://rd.yahoo.com/M=233467.2381293.3996454.2316829/D=yahoo_top/
S=2716149:M3/A=1311104/R=0/O=1/I=yahoo-fp-marketplace/overstock-watches/*http://shop.store.yahoo.com/cgi-
bin/clink?overstock3+shopping:dmfp.marketplace/overstock-watches+http://us.rmi.yahoo.com/rmi/http://www.overstock.com/rmi-
framed-url/http://www.overstock.com/cgi-bin/d2.cgi%3FPAGE=STATICPAGE%26PAGE_ID=406%26cid=24184%26fp=f

0.002395487418086108

http://www.gojane.com/who.html

0.05012596598835563

Level Eight

http://www.yahoo.com

6.6122977991905175E-9

http://www.gojane.com/sizechartpage.html

0.22603949758375205

http://www.gojane.com/louietank.html

0.04455190069444609

http://www.gojane.com/longsleeves.html

0.04455190069444609

http://www.gojane.com/jeans.html

0.04455190069444609

Appendix 3 The New Results

A-1 “Automobile Manufacturer” Case (ES)

Level One

http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1620

0.4223452874385104

http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/
Manufacturers_and_Suppliers/

0.00965370328620864

http://dir.yahoo.com/Regional/Countries/Canada/Business_and_Economy/Business_to_Business/
Automotive/Industry_Information/Manufacturers_and_Suppliers/

0.00826283875052485

<http://www.japanauto.com/>

0.00588277876957169

<http://www.aiam.org/>

0.00468378645210461

<http://yp.yahoo.com/py/ypResults.py?stx=manufacturer+automobile>

0.00239185455486392

Level Two:

<http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile>

0.49195917773042336

<http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile>

0.08212736856281715

<http://yp.search.yahoo.com/search/ypredirect?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile>

0.03938170883459406

http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D

0.0038421179350692104

http://story.news.yahoo.com/news?tmpl=story&u=/nm/20030415/pl_nm/environment_diesel_dc_2

0.001121058967534602

http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645

0.000821693780163709

Level Three

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DNWT/*-
<http://search.yahoo.com/search?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.6493573501509173

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DPRT/*-
http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b&pref_cancel=http%3a//search.yahoo.com/search/dir%3fhc=33%26hs=1645%26ei=UTF-8%26p=manufacturer%2bautomobile%2b

0.002246184929257296

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DAST/*-
<http://search.yahoo.com/dir/advanced?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile+>

0.00140386558078581

Level Four

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WNDT/*-
<http://search.yahoo.com/search/dir?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.2627204741579596
http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WPRT/*-
http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b&p_ref_cancel=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645%26ei=UTF-8
0.0015764438741177884

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WAST/*-
<http://search.yahoo.com/web/advanced?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.001379388389853065

Level Five

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DNNT/*-
<http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.4056272344389177

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DPRT/*-
http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b&p_ref_cancel=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645%26ei=UTF-8

0.006447506810988319

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DAST/*-
<http://search.yahoo.com/dir/advanced?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.004029691756867699

Level Six

<http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile+>

0.4441735461571298

<http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile+>

0.03328261338589

<http://yp.search.yahoo.com/search/ypredirect?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile+>

0.015960083611947253
http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645%26ei%3DUTF-8

0.0014901608802347372

http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645%26ei%3DUTF-8

0.0011176206601760529

Level Seven

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DNWT/*-
<http://search.yahoo.com/search?p=manufacturer+automobile+&hc=33&hs=1646&ei=UTF-8>

0.6493573501509167

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DPRT/*-
http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b&p_ref_cancel=http%3a//search.yahoo.com/search/dir%3fhc=33%26hs=1646%26ei=UTF-8%26p=manufacturer%2bautomobile%2b

0.0022461849292572944

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DAST/*-
<http://search.yahoo.com/dir/advanced?hc=33&hs=1646&ei=UTF-8&p=manufacturer+automobile+>

0.001403865580785809

Level Eight

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WNDT/*-
<http://search.yahoo.com/search/dir?p=manufacturer+automobile+&hc=33&hs=1646&ei=UTF-8>

0.2627204741579596

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WPRT/*-
http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b&pref_cancel=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1646%26ei=UTF-8

0.0015764438741177884

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WAST/*-
<http://search.yahoo.com/web/advanced?p=manufacturer+automobile+&hc=33&hs=1646&ei=UTF-8>

0.001379388389853065

A-2 “Automobile Manufacturer” Case (HITS)

Level One

<http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1620>

0.4223452874385104

http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/

0.00965370328620864

http://dir.yahoo.com/Regional/Countries/Canada/Business_and_Economy/Business_to_Business/Automotive/Industry_Information/Manufacturers_and_Suppliers/

0.00826283875052485

<http://www.japanauto.com/>

0.00588277876957169

<http://www.aiam.org/>

0.00468378645210461

<http://yp.yahoo.com/py/ypResults.py?stx=manufacturer+automobile>

0.00239185455486392

Level Two

[.http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile](http://search.yahoo.com/search/dir?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile)

0.21442 164009203613

[.http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile](http://img.search.yahoo.com/search/images?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile)

0.19381539445954882

<http://yp.search.yahoo.com/search/ypredirect?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile>
0.08975808754671673

http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645
0.008756886589923582

http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dmanufacturer%2Bautomobile%2B%26hc%3D33%26hs%3D1645

0.006567664942442686

http://search.news.yahoo.com/search/news/?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile+&c=news_photos

0.039810864163511435

Level Three

[http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.news.yahoo.com/search/news/%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645&.src=yn&.v=0&.u=4prq2t0vabv11&partner=&.p=&promo=&.last="](http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.news.yahoo.com/search/news/%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645&.src=yn&.v=0&.u=4prq2t0vabv11&partner=&.p=&promo=&.last=)

7.92835205282227E-5

[http://edit.yahoo.com/config/eval_forgot_pw?new=1&.done=http://search.news.yahoo.com/search/news/?p=manufacturer+automobile+&hc=33&hs=1645&.src=yn&partner=&.intl=us&pkg=&stepid=&.last="](http://edit.yahoo.com/config/eval_forgot_pw?new=1&.done=http://search.news.yahoo.com/search/news/?p=manufacturer+automobile+&hc=33&hs=1645&.src=yn&partner=&.intl=us&pkg=&stepid=&.last=)

2.378505615846681E-4

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DAST/*-
<http://search.yahoo.com/dir/advanced?hc=33&hs=1645&ei=UTF-8&p=manufacturer+automobile+>

0.006569256132473994

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DPRT/*-
http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645%26ei=UTF-8%26p=manufacturer%2bautomobile%2b%26cancel=http%3a//search.yahoo.com/search/dir%3fhc=33%26hs=1645%26ei=UTF-8%26p=manufacturer%2bautomobile%2b%26

0.010510809811958392

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DNWT/*-
<http://search.yahoo.com/search?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.22534729026338482

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DNNT/*-
<http://search.news.yahoo.com/search/news?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.0486114094234305

Level Four

<http://login.yahoo.com/config/login?.src=yn&partner=&.v=0&.u=4prq2t0vabv11&.intl=us&.done=http%3a//search.news.yahoo.com/search/news/%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645>

1.9959365254347037E-4

[http://login.yahoo.com/config/login?new=1&.intl=us&.done=http%3a//search.news.yahoo.com/search/news/%3fp=manufacturer+automobile&.src=yn&referrer=&partner="](http://login.yahoo.com/config/login?new=1&.intl=us&.done=http%3a//search.news.yahoo.com/search/news/%3fp=manufacturer+automobile&.src=yn&referrer=&partner=)
1.4299779307903158E-4

http://login.yahoo.com/config/login?.src=srch&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search/dir%253fp=manufacturer%252bautomobile%252b%26pref_cancel=http%253a//search.yahoo.com/search/dir%253fhc=33%2526hs=1645%2526ei=UTF-8%2526p=manufacturer%252bautomobile%252b

4.370338289909101E-4

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WAST/*-
<http://search.yahoo.com/web/advanced?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.0075768202961826025

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WPRT/*-
http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b&pref_cancel=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645%26ei=UTF-8

0.00865922319563726

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=WNDT/*-
<http://search.yahoo.com/search/dir?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.11243843257484636

Level Five

http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.news.yahoo.com/search/news/%3fp=manufacturer+automobile&.src=yn&.v=0&.u=0bfgfrovabv2g&partner=&.p=&promo=&.last=

1.2982932358439946E-4

http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search/dir%253fp=manufacturer%252bautomobile%252b%26pref_cancel=http%253a//search.yahoo.com/search/dir%253fhc=33%2526hs=1645%2526ei=UTF-8%2526p=manufacturer%252bautomobile%252b&.src=srch&.v=0&.u=aqlkb14vabv2i&partner=&.p=&promo=&.last=

3.38466244138035E-4

http://edit.yahoo.com/config/eval_forgot_pw?new=1&.done=http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b&pref_cancel=http%3a//search.yahoo.com/search/dir%3fhc=33%26hs=1645%26ei=UTF-8%26p=manufacturer%2bautomobile%2b&.src=srch&partner=&.intl=us&pkg=&stepid=&.last=

5.076993662070525E-4

http://login.yahoo.com/config/login?.src=srch&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search%253fp=manufacturer%252bautomobile%252b%26pref_cancel=http%253a//search.yahoo.com/search%253fp=manufacturer%252bautomobile%252b%2526hc=33%2526hs=1645%2526ei=UTF-8

0.0010386345886751957

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DAST/*-
<http://search.yahoo.com/dir/advanced?p=manufacturer+automobile+&hc=33&hs=1645&ei=UTF-8>

0.012527337920132253

http://srd.yahoo.com/S=2766679/K=manufacturer+automobile+/v=2/l=DPRT/*-
http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b&pref_cancel=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645%26ei=UTF-8

0.020043740672211603

Level Six

<http://login.yahoo.com/config/login?.src=yn&partner=&.v=0&.u=0bfgfrovabv2g&.intl=us&.done=http%3a//search.news.yahoo.com/search/news/%3fp=manufacturer+automobile>
0.017611576701260245

http://login.yahoo.com/config/login?.src=srch&partner=&.v=0&.u=aqlkb14vabv2i&.intl=us&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search/dir%253fp=manufacturer%252bautomobile%252b%26pref_cancel=http%253a//search.yahoo.com/search/dir%253fhc=33%2526hs=1645%2526ei=UTF-8%2526p=manufacturer%252bautomobile%252b

0.03522315340252049

http://login.yahoo.com/config/login?new=1&.intl=us&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b&.src=srch&referer=&partner=0.023810527381132797

http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search%253fp=manufacturer%252bautomobile%252b%26pref_cancel=http%253a//search.yahoo.com/search%253fp=manufacturer%252bautomobile%252b%2526hc=33%2526hs=1645%2526ei=UTF-8&.src=srch&.v=0&.u=ckfkh1svabv39&partner=&.p=&promo=&.last=

0.048767541093649856

http://edit.yahoo.com/config/eval_forgot_pw?new=1&.done=http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b&pref_cancel=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645%26ei=UTF-8&.src=srch&partner=&.intl=us&pkg=&stepid=&.last=

0.07326314610115682

http://login.yahoo.com/config/login?.src=srch&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search/dir%253fp=manufacturer%252bautomobile%252b%2526hc=33%2526hs=1645%2526ei=UTF-8

0.07111629358546213

Level Seven

http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b&.src=srch&.v=0&.u=7rc913kvabv4g&partner=&.p=&promo=&.last=

0.003571522160068401

http://edit.yahoo.com/config/eval_forgot_pw?new=1&.done=http://search.yahoo.com/search/preferences?pref_done=http://search.yahoo.com/search/dir?p=manufacturer+automobile+&.src=srch&partner=&.intl=us&pkg=&stepid=&.last=

0.03271856876454578

http://login.yahoo.com/config/login?.src=srch&partner=&.v=0&.u=ckfkh1svabv39&.intl=us&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search%253fp=manufacturer%252bautomobile%252b%26pref_cancel=http%253a//search.yahoo.com/search%253fp=manufacturer%252bautomobile%252b%2526hc=33%2526hs=1645%2526ei=UTF-8

0.020873674710010694

http://login.yahoo.com/config/login?new=1&.intl=us&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b&.src=srch&referer=&partner=

0.3036542238772693

http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search/dir%253fp=manufacturer%252bautomobile%252b%26pref_cancel=http%253a//search.yahoo.com/search/dir%253fp=manufacturer%252bautomobile%252b%2526hc=33%2526hs=1645%2526ei=UTF-8&.src=srch&.v=0&.u=2fglolsvabv6k&partner=&.p=&promo=&.last=

0.007143044320136802

http://edit.yahoo.com/config/eval_forgot_pw?new=1&.done=http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b&pref_cancel=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b%26hc=33%26hs=1645%26ei=UTF-8&.src=srch&partner=&.intl=us&pkg=&stepid=&.last=

0.05054648165629621

Level Eight

http://login.yahoo.com/config/login?.src=srch&partner=&.v=0&.u=7rc913kvabv4g&.intl=us&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer%2bautomobile%2b

0.014782749981403104

http://login.yahoo.com/config/login?new=1&.intl=us&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%3a//search.yahoo.com/search/dir%3fp=manufacturer+automobile&.src=srch&referer=&partner=

0.13616758543876784

http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%3a//search.yahoo.com/search%3fp=manufacturer%2bautomobile%2b&.src=srch&.v=0&.u=5a8vv08vabv7h&partner=&.p=&promo=&.last=

0.006290860879379372

http://edit.yahoo.com/config/eval_forgot_pw?new=1&.done=http://search.yahoo.com/search/preferences?pref_done=http://search.yahoo.com/search?p=manufacturer+automobile+&.src=srch&partner=&.intl=us&pkg=&stepid=&.last=

0.04875971127088949

http://login.yahoo.com/config/login?.src=srch&partner=&.v=0&.u=2fglolsvabv6k&.intl=us&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%253a//search.yahoo.com/search/dir%253fp=manufacturer%252bautomobile%252b%26pref_cancel=http%253a//search.yahoo.com/search/dir%253fp=manufacturer%252bautomobile%252b%2526hc=33%2526hs=1645%2526ei=UTF-8

0.02956549996280621

http://edit.yahoo.com/config/eval_register?.intl=us&new=1&.done=http%3a//search.news.yahoo.com/search/news/%3fp=manufacturer+automobile&.src=yn&.v=0&.u=0bfgfrovabv2g&partner=&.p=&promo=&.last=

0.0010175227247058956

B-1 “Java Developer” Case (ES)

Level One

<http://www.ibm.com/developerworks/java/>

0.4128629605998993

<http://java.sun.com/>

0.03965328753334086

<http://developer.java.sun.com>

0.02427424567471161

<http://www.javalobby.org/>

0.01184420477709548

<http://www.jfind.com/javadev.shtml>

0.00771643412249827

<http://www.javaalmanac.com/>

0.00393804118843172

Level Two

<http://www-106.ibm.com/developerworks/theme/>

0.21389528634262828

<http://www-106.ibm.com/developerworks/java/library/l-secjav.html>

0.20174847712950947

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.01364113199828978

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.07649925899708516

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www7b.boulder.ibm.com/wsdd/&origin=j>

0.0226409369130074

<http://www.symbian.com/news/2002/expo03-feat.html>

0.00287951764014932

Level Three

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/?origin=dwheader>

0.001210995453421637

<http://www-106.ibm.com/developerworks/search/help-dw.html>

0.06146663549059433

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/ibm/?origin=dwheader>

0.0014597188413343466

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/privacy/?origin=dwheader>

0.0014597188413343466

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/legal/?origin=dwheader>

0.0014597188413343466

<http://www-106.ibm.com/developerWorks/ibm>

0.21099918292235487

Level Four

<http://www-106.ibm.com/developerworks/ibm/library/i-jmsmq/>

0.23716478378508452

<http://www-106.ibm.com/developerworks/ibm/library/i-voicestudio>

0.11917806959255715

<http://www-106.ibm.com/developerworks/java/library/j-newviews/>

0.09861675824510517

<http://www-106.ibm.com/developerworks/ibm/library/i-gctroub/>

0.08488879062305194

http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7b.software.ibm.com/wsdd/library/techarticles/0304_yu/you1.html&origin=devdomain

0.004516669057178373

<http://advisorevents.com/event/CIW0306>

0.000012980126743490

Level Five

<http://www.ibm.com/developerworks/ibm/>

0.1695241698584679

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www-106.ibm.com/developerworks/newsletter/&origin=dw-article>

0.15323753370696858

<http://www-106.ibm.com/developerworks/ibm/>

0.17957025766079823

<http://www-106.ibm.com/developerworks/>

0.08825029662040426

<http://www-106.ibm.com/developerworks/ibm/library/i-garbage1/>

0.060125176060825486

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.006904296214092955

Level Six

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/?origin=dwheader>

0.00077209328084698

<http://www-106.ibm.com/developerworks/search/help-dw.html>

0.03940105770177632

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/ibm/?origin=dwheader>

0.0009\0669902034472

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/privacy/?origin=dwheader>

0.00066990203447264

<http://www-106.ibm.com/developerworks/newsletter/>

0.16806404906393016

<http://www.ibm.com/developerworks/edu/i-dw-i-metro7-i.html>

0.021696517319191425

Level Seven

<http://www-106.ibm.com/developerworks/feedback/>

0.017999224037587597

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/legal/?origin=dwheader>

0.0016615912959097184

<http://www-106.ibm.com/developerworks/sitemap.html>

0.1413875600951131

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.13521957543807514

http://www.ibm.com/partnerworld/pwhome.nsf/weblook/nws_ltr.html

0.005280106009190748

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www-3.ibm.com/software/mailling-lists/newsletter/>

0.000973917027345433

Level Eight

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.012308360763024134

<http://www-106.ibm.com/developerWorks/ibm>

0.22125817215535068

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.lotus.com/lld&origin=devdomain>

0.028663843086830794

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www-1.ibm.com/servers/esdd/&origin=devdomain>

0.08956975018068146

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www-106.ibm.com/developerworks/tivoli/&origin=devdomain>

0.07216359118022049

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/websphere/developer&origin=devdomain>

0.03605780472382218

B-2 “Java Developer” Case (HITS)

Level One

<http://www.ibm.com/developerworks/java/>

0.4128629605998993

<http://java.sun.com/>

0.03965328753334086

<http://developer.java.sun.com>

0.02427424567471161

<http://www.javalobby.org/>

0.01184420477709548

<http://www.jfind.com/javadev.shtml>

0.00771643412249827

<http://www.javaalmanac.com/>

0.00393804118843172

Level Two

<http://www-106.ibm.com/developerworks/java/library/j-jsp04153.html>

0.12385042367469752

<http://www-106.ibm.com/developerworks/java/library/j-jstl0415/>

0.0816320479604754

<http://www-106.ibm.com/developerworks/java/library/j-newviews/>

0.048788759036143776

<http://www-106.ibm.com/developerworks/java/library/j-jtp04223.html>

0.04160064859367833

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.039942197195027083

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.0028765258598279033

Level Three

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7.software.ibm.com/vad.nsf/&origin=devdomain>

0.0058784900539847335

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7b.software.ibm.com/dmdd/&origin=devdomain>

0.013169523941762296

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/?origin=dwheader>

8.013063312054434E-4

<http://www-106.ibm.com/developerworks/search/help-dw.html>

0.03923949001771501

<http://www-106.ibm.com/developerWorks/ibm>

0.10671560174216498

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/websphere/developer&origin=devdomain>

0.013438694555219382

Level Four

<http://www.ibm.com/>

0.000205964115483258

<http://www.ibm.com/developerworks/ibm/library/i-db2zope/>

0.020039621351926173

<http://www.ibm.com/developerworks/ibm/library/i-toolbox/>

0.06564104260113589

<http://www.ibm.com/developerWorks/grid?ca=grid-db2dd>

0.11143644175165106

<http://www-106.ibm.com/developerworks/newsletter/?loc=db2dd>

0.1821211204743337

<http://www.ibm.com/contact/employees/us/>

0.00024088683261459

Level Five

<http://www.ibm.com/developerworks/toolbox/?ca=dtb-ibmcom>

0.029567547277975743

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/&origin=dwheader>

0.0038109227927482243

<http://www-106.ibm.com/developerworks/>

0.06674839272599699

<http://www-106.ibm.com/developerworks/ibm/>

0.07877688246828192

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www-106.ibm.com/developerworks/newsletter/&origin=dw-article>

0.08927879326156683

<http://www-106.ibm.com/developerworks/toolbox/>

0.06459539222818152

Level Six

<http://www-106.ibm.com/developerworks/ibm/nav.html>

0.03258322336703579

<http://www-106.ibm.com/developerworks/search/help-dw.html>

0.024325429742398142

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/&origin=dwheader>

5.636098154430383E-4

<http://www6.software.ibm.com/devcon/devcon/docs/contents.html>

0.022717969537488034

<http://www-106.ibm.com/developerworks/journal/>

0.04810019809602374

<http://www-106.ibm.com/developerworks/newsletter/>

0.05950923478994232

Level Seven

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7.software.ibm.com/vad.nsf/&origin=devdomain>

0.00792902808842444

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7b.software.ibm.com/dmdd/&origin=devdomain>

0.01731433559770713

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/contact/?origin=dwheader>

0.0010078422372957762

<http://www-106.ibm.com/developerWorks/ibm>

0.08741250004725039

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=www-1.ibm.com/servers/esdd/&origin=devdomain>

0.024955961436401802

<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.ibm.com/websphere/developer&origin=devdomain>

0.01830308593520172

Level Eight

<http://www.ibm.com/>

0.000222792621737222

<http://www.ibm.com/developerworks/ibm/library/i-db2zope/>

0.020123456985045775

<http://www.ibm.com/developerworks/ibm/library/i-toolbox/>

0.05999386569602476

<http://www.ibm.com/developerWorks/grid?ca=grid-db2dd>

0.11962546196543437

<http://www-106.ibm.com/developerworks/newsletter/?loc=db2dd>

0.1859766203652267

<http://www.ibm.com/contact/employees/us/>

0.00026330248250991033

C-1 “Artificial Intelligence” Case (ES)

Level One

http://dir.yahoo.com/Entertainment/Movies_and_Film/Titles/Science_Fiction_and_Fantasy/Artificial_Life_Forms/A_I__Artificial_Intelligence/

0.1886505673185579

http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/

0.18073010550157712

http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/European_Coordinating_Committee_for_Artificial_Intelligence__ECCAI_/

0.07655744993755742

<http://www.ai.mit.edu/>

0.002408330522024467

http://www1.overture.com/d/sr/?xargs=02u3hs9yoaj00SvTCChBDFXx0qBc45AxSKk%2FLC2ogXaCEwZghg%2BMxq9gZg2q93Yt
vvL3nCm83geDEa1UU%2Fptsyf4BZGiVRMJjYzDF2HxQAZBbqIQ5QUWQOQBfKLGcLeLDHo8w78Z1jNNiAsml1pWTJcFJC
vAjUc0DkRmkCtEuDf4i9j2as76q3WD9MG9vacb3E3rKGJUFFqxDOpqlP9Dh23%2BPZwBk%2B1ICNknmylOc7%2BBaABV

0.00124 25659234176027

http://www.aaai.org/

0.0010729828466247205

Level Two

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=42779322/*http://www.rottentomatoes.com/movie-10000163/about.php

0.3581355972297218

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=27550374/*http://www.dreamworksfansite.com/ai/

0.00632 6540459276429

http://srd.yahoo.com/S=245960:D1/CS=245960/SS=42656828/*http://www.singinst.org/

0.006093600122767582

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=58521013/*http://www.filmtracks.com/titles/ai.html

0.003017693639517619

http://srd.yahoo.com/S=245960:D1/CS=245960/SS=245962/*http://www.aaai.org/

0.0018874133144593935

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=39249876/*http://aimovie.warnerbros.com/

0.0012 16582316065081

Level Three

http://www.qksrv.net/click-637882-
5463217?loc=http%3A//search.ebay.com/search/search.dll%3FMfcISAPICCommand%3DGetResult%26ht%3D1%26SortProperty%3D
MetaEndSort%26query%3Dartificial+intelligence

0.6249654029432038

http://www.aaai.org/AITopics/html/current.html

0.015821680547240133

http://www.amazon.com/exec/obidos/ASIN/0312280610/o/qid=990202631/sr=8-1/ref=aps_sr_b_1_1/stevenspielbefan/107-3769883-
5568548

0.0005077918791334506

http://www.filmtracks.com/ratings/

0.00021974556315611295

http://www.aimovie.com/

0.00021974556315611295

Level Four

<http://cgi1.ebay.com/aw-cgi/eBayISAPI.dll?AddSavedSearch&savedsearch=http://search.ebay.com/search/search.dll?GetResult%26st%3d2%26SortProperty%3dMetaEndSort%26query%3Dartificial%2Bintelligence%26ht%3D1%26combine%3Dy>

0.021350212346402487
<http://search-completed.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2&SortProperty=MetaEndSort>

0.32687508397495735

<http://search.stores.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&rd=on&fp=4>

0.11469190429777457

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3321736999&category=2296>

0.0073994776966306176

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1>

0.004932985131087078

Level Five

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2>

0.28708309373143287
<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3319389028&category=2296>

0.000680305224302247

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3320001913&category=2296>

0.000583118763687640

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=2520292489&category=1572>

0.000583118763687640

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1&rflsActive=0>

0.000380115450536721

Level Six

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3515147401&category=29373>

0.06702481550093764

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3127707672&category=232>

0.056270233524255794

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=2524047773&category=29940>

0.04691737085065635
<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3322105378&category=2307>

0.04021488930056259

<http://cgi1.ebay.com/aw-cgi/eBayISAPI.dll?AddSavedSearch&savedsearch=http://search.ebay.com/search/search.dll?GetResult%26st%3d2%26query%3Dartificial%2Bintelligence%26ht%3D1%26combine%3Dy>

0.030804922058411757

Level Seven

http://rd.yahoo.com/search/navbar/top/goo/intelligence+artificial/*http://search.yahoo.com/search?n=20&p=intelligence+artificial&h=w&hc=6&hs=258

0.10846724050311812

http://rd.yahoo.com/search/navbar/nextc/*http://search.yahoo.com/search/dir?n=20&p=intelligence+artificial&hc=6&hs=258&b=4&h=C&xargs=

0.03109231321506841

http://rd.yahoo.com/search/navbar/top/news/intelligence+artificial/*http://search.news.yahoo.com/search/news?p=intelligence+artificial&hc=6&hs=258

0.01284202253426246

<http://search.yahoo.com/search/options?n=20&p=intelligence+artificial&h=c>

0.01139843746573754

Level Eight

http://srd.yahoo.com/S=2766679:WS0/R=1/K=intelligence+artificial/CS=245765/*http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/

0.3078540975821424

http://srd.yahoo.com/S=2766679:WS0/R=2/K=intelligence+artificial/CS=519794/*http://dir.yahoo.com/Recreation/Games/Computer_Games/Internet_Games/Web_Games/Artificial_Intelligence/

0.20396652565076481

http://srd.yahoo.com/S=2766679:WS0/R=3/K=intelligence+artificial/CS=21390/*http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Computers/Software/Artificial_Intelligence/

0.09435562493861237

<http://search.yahoo.com/search/options?n=20&p=intelligence+artificial&h=w&hc=6&hs=258>

0.03942384946138763

C-2 “Artificial Intelligence” Case (HITS)

Level One

http://dir.yahoo.com/Entertainment/Movies_and_Film/Titles/Science_Fiction_and_Fantasy/Artificial_Life_Forms/A_I__Artificial_Intelligence/

0.1886505673185579

http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/

0.18073010550157712

http://dir.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Organizations/European_Coordinating_Committee_for_Artificial_Intelligence__ECCAI/

0.07655744993755742

<http://www.ai.mit.edu/>

0.002408330522024467

<http://www1.overture.com/d/sr/?xargs=02u3hs9yoaj00SvTCCbBDFXx0qBc45AxSKk%2FLC2ogXaCEwZghg%2BMxq9gZg2q93Yt vvL3nCm83geDEa1UU%2Fptsyf4BZGiVRMJjbbYzDF2HxQAZBbqJQ5QUWQOQBfKLGcLeLDHo8w78Z1jNNiAsml1pWTJcFJC vAjUc0DkRmkCtEuDf419j2as76q3WD9MG9vacb3E3rKGJUffqxDOpqlP9Dh23%2BPZwBk%2B11CNkmylOc7%2BBaABV>

0.00124 25659234176027

<http://www.aaai.org/>

0.0010729828466247205

Level Two

<http://www.aaai.org/AITopics/html/current.html>

0.3727713512640388

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=42779322/*http://www.rottentomatoes.com/movie-10000163/about.php

0.0622171040374019

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=27550374/*http://www.dreamworksfansite.com/ai/

0.0061134539539702375

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=58521013/*http://www.filmtracks.com/titles/ai.html

0.004075635969313492

http://srd.yahoo.com/S=245960:D1/CS=245960/SS=245962/*http://www.aaai.org/

0.0037054488626334244

http://srd.yahoo.com/S=27550299:D1/CS=27550299/SS=39249876/*http://aimovie.warnerbros.com/

0.002037817984656746

Level Three

http://www.amazon.com/exec/obidos/ASIN/0312280610/o/qid=990202631/sr=8-1/ref=aps_sr_b_l_1/stevenspielbefan/107-3769883-5568548

0.0004242441872483609

<http://www.filmtracks.com/ratings/>

0.0002624270820026333

<http://www.aimovie.com/>

0.0001266409656768125

http://www.boston.com/dailyglobe2/112/science/War_games%2B.shtml

0.006717743375608712

<http://www.e4engineering.com/item.asp?id=48525&type=News&pub=eng>

0.005013518259290454

<http://news-service.stanford.edu/news/2003/april23/engelmorobit-423.html>

0.020153230126826134

Level Four

<http://www.qksrv.net/click-637882-5463217?loc=http%3A//search.ebay.com/search/search.dll%3FMcISAPICommand%3DGetResult%26ht%3D1%26SortProperty%3DMetaEndSort%26query%3Dartificial+intelligence>

0.5001654392592857

http://www.signonsandiego.com/news/uniontrib/mon/business/news_mz1b21pop.html

0.00015467627028595

<http://www.post-gazette.com/healthscience/20030421scienotessci8p8.asp>

0.00015467627028595

http://www.boston.com/dailyglobe2/110/living/Tired_of_vacuuming_Have_a_200_robot_it_instead%2B.shtml

0.00030086920381002

<http://www.eedesign.com/news/OEG20030419S0001>

0.00020885014960929

<http://www.madison.com/captimes/books/topic/nonfiction/47239.php>

0.000175467627028595

Level Five

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1>

0.0031570795879230754

<http://search-completed.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2&SortProperty=MetaEndSort>

0.14101535626624515

<http://cgi1.ebay.com/aw-cgi/eBayISAPI.dll?AddSavedSearch&savedsearch=http://search.ebay.com/search/search.dll?GetResult%26st%3d2%26SortProperty%3dMetaEndSort%26query%3Dartificial%2Bintelligence%26ht%3D1%26combine%3Dy>

0.006393555351401199

<http://search.stores.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&rd=on&fp=4>

0.08238442364230054

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3321736999&category=2296>

0.0054703215243748545

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=2524047773&category=29940>

0.0055282686755098575

Level Six

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1&rflsActive=0>

0.0027778434223966753

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2>

0.13705082311863293

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3320001913&category=2296>

0.011305699452895644

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3319389028&category=2296>

0.013196227055445455

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=2520292489&category=1572>

0.011305699452895644

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3318919029&category=197>

0.0037685664842985484

Level Seven

<http://cgi1.ebay.com/aw-cgi/eBayISAPI.dll?AddSavedSearch&savedsearch=http://search.ebay.com/search/search.dll?GetResult%26st%3d2%26query%3Dartificial%2Bintelligence%26ht%3D1%26combine%3Dy>

0.010353211217011249

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3127707672&category=232>

0.010353586877930887

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3515147401&category=29373>

0.0129419789220314

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3322105378&category=2307>

0.0077651878730632526

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3321753241&category=2298>

0.005176789735183612

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3321759561&category=2307>

0.0077655227336990915

Level Eight

<http://search.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&SortProperty=MetaEndSort&st=2&rp=1>

1.0958794412941257E-4

<http://search-completed.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&ht=1&combine=y&st=2&SortProperty=MetaEndSort>

0.26610218765690125

<http://search.stores.ebay.com/search/search.dll?GetResult&query=artificial+intelligence&rd=on&fp=4>

0.19603150965058178

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3321736999&category=2296>

0.013930856742803923

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=2524047773&category=29940>

0.0024404728700657954

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3514721637&category=1107>
0.0013920736363581022

D-1 “WWW conference” Case (ES)

Level One

<http://search.news.yahoo.com/search/news?p=conference+www+&c=news>

0.30751 61142110977

<http://www.sys-con.com/webwest/>

0.11375 166226648216

<http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=31>

0.089570527213745507

<http://www.iisg.nl/~w3vl/>

0.01277 8443675237054

<http://www.www2002.org/>

0.00996 7715504353978

<http://www oulu.fi/~spaceweb/lib/>

0.00704 1720594491317

Level Two

<http://search.yahoo.com/search/dir?c=news&ei=UTF-8&p=conference+www>

0.44631382108011247

<http://yp.search.yahoo.com/search/ypredirect?c=news&ei=UTF-8&p=conference+www>

0.08895520277167124

http://rd.yahoo.com/dailynews/front/welcome/signin/*http://edit.yahoo.com/config/login?.src=yn&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dconference%2Bwww%2B%26c%3Dnews

0.010107621002417598

http://rd.yahoo.com/dailynews/front/welcome/cst/*http://edit.news.yahoo.com/news?tmpl=s&u=/static/customize/set_content&.done=http%3A%2F%2Fsearch.news.yahoo.com%2Fsearch%2Fnews%2F%3Fp%3Dconference%2Bwww%2B%26c%3Dnews

0.006765269972817512

http://rd.yahoo.com/M=224039.1983422.3465437.2905688/D=news/P=m2n1l1cf13000500/S=53720272:HEAD/A=1030394/R=0/*http://news.yahoo.com

0.005436305193304572

http://rd.yahoo.com/M=243098.2885920.4233620.2839171/D=news/S=53720272:PB/A=1354070/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=243098.2885920.4233620.2839171/D=news/S=53720272:PB/A=1354070/R=5/1051061649+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/poweredby/smb/index-yahoo.html

0.0010872610386609145

Level Three

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=DNWT/*-http://search.yahoo.com/search?p=conference+www+c=news&ei=UTF-8

0.522263057132637

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=DPRT/*-http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search/dir%3fp=conference%2bwww%2b&pref_cancel=http%3a//search.yahoo.com/search/dir%3fc=news%26ei=UTF-8%26p=conference%2bwww%2b

0.015382706520013337

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=DAST/*-http://search.yahoo.com/dir/advanced?c=news&ei=UTF-8&p=conference+www+

0.00992811611974855

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=DYH/*-http://www.yahoo.com

0.00920845331237966

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=DSH/*-http://search.yahoo.com

0.00544837132119307

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=DHL/*-http://help.yahoo.com/help/us/ysearch/

0.00544837132119307

Level Four

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=WPRT/*-http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search%3fp=conference%2bwww%2b&pref_cancel=http%3a//search.yahoo.com/search%3fp=conference%2bwww%2b%26c=news%26ei=UTF-8

0.042413424562309614

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=WAST/*-http://search.yahoo.com/web/advanced?p=conference+www+&c=news&ei=UTF-8

0.039440922610081024

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=WYH/*-http://www.yahoo.com

0.027184810338212673

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=WHL/*-http://help.yahoo.com/help/us/ysearch/

0.002070378771614886

http://rd.yahoo.com/M=243098.2885926.4233627.1288602/D=yahoo/S=2766679/PB/A=1496187/R=0/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=243098.2885926.4233627.1288602/D=yahoo/S=2766679/PB/A=1496187/R=1/1051061823+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/poweredby/smb/index-yahoo.html

0.002070378771614886

http://srd.yahoo.com/S=2766679/K=conference+www+/v=2/l=WSH/*-http://search.yahoo.com

0.002070378771614886

Level Five

http://rd.yahoo.com/M=224039.1984929.3466713.1922510/D=yahoo/P=f2f06akd11000900/S=2766679:HEAD/A=1030697/R=0/*http://www.yahoo.com

0.014838840544799423

http://rd.yahoo.com/search/navbar/nests/*http://search.yahoo.com/search?p=conference+www&b=21&hc=1&hs=1&xargs=0

0.4292790609211374

http://srd.yahoo.com/S=2766679:WS1/R=2/K=conference+www/SS=2243065/OCS=253475/T=1037831084/F=98c5d375e450945a65eb944ddb25379c/*http://www.conference-board.org/

0.005813006178362968

<http://search.yahoo.com/search/options?p=conference+www&h=w&hc=1&hs=27>

0.0013926313345262416

http://srd.yahoo.com/S=2766679:WS1/R=1/K=conference+www/SS=139794/OCS=135823/T=1037831084/F=e7b0a0de0bdd169766128387d2f854e1/*http://www.ncsl.org/

0.0013926313345262416

http://rd.yahoo.com/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=4/id=noscript/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=179842.1351601.3214785.1288602/D=yahoo/S=2766679:PB/A=1168887/R=5/1037831084+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/yahoo_poweredby/
0.0002.635575170681213

Level Six

http://rd.yahoo.com/M=247604.3199797.4508202.2586395/D=yahoo_top/S=2716149:M3/A=1549176/R=0/*http://www.lendingtree.com/borrower/alliance/from.asp?source=11220&esourceid=11220&whereto=bannerrealtor&partner=yahoo

0.1904872435510694

http://rd.yahoo.com/M=251707.3199787.4508186.2584085/D=yahoo_top/S=2716149:M/A=1549115/R=1/*http://www.coastalcontacts.com/referpages/refer_page121.htm

0.06144085035494264

http://rd.yahoo.com/M=251707.3199787.4508186.2584085/D=yahoo_top/S=2716149:M/A=1549115/R=0/*http://www.coastalcontacts.com/referpages/refer_page121.htm

0.06144085035494248

http://rd.yahoo.com/M=243098.2885930.4233629.3055070/D=yahoo_top/S=2716149:PB/A=1354075/R=0/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=243098.2885930.4233629.3055070/D=yahoo_top/S=2716149:PB/A=1354075/R=1/1051061888+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/poweredby/consumer/index-yahoo.html

0.034101627729752344

http://rd.yahoo.com/M=243098.2885930.4233629.3055070/D=yahoo_top/S=2716149:PB/A=1354075/R=0/*http://shop.store.yahoo.com/cgi-bin/clink?hp3+shopping:dmad/M=243098.2885930.4233629.3055070/D=yahoo_top/S=2716149:PB/A=1354075/R=1/1051061891+http://us.rmi.yahoo.com/rmi/http://www.compaq.com/rmi-framed-url/http://www.compaq.com/bridge/poweredby/consumer/index-yahoo.html

0.03410162772975225

http://edit.yahoo.com/config/eval_forgot_pw?new=1&.done=http://search.yahoo.com/search/preferences?pref_done=http%3a//search.yahoo.com/search%3fp=conference%2bwww%2b&pref_cancel=http%3a//search.yahoo.com/search%3fp=conference%2bwww%2b%26c=news%26ei=UTF-8&.src=srch&partner=&.intl=us&pkg=&stepid=&.last=

0.020661787337124458

Level Seven

<http://www.lendingtree.com/stm/offers/realty.asp?realtysource=11220&partner=yahoo>

0.3871557504304019

http://www.coastalcontacts.com/index_Yahoo.cfm
0.03293624449039486

<http://www.yahoo.com>

0.009786079102244806

http://login.yahoo.com/config/login?new=1&.intl=us&.done=http%3a//search.yahoo.com/search/preferences%3fpref_done=http%3a//search.yahoo.com/search%3fp=conference%2bwww%2b&.src=srch&referer=&partner=

0.0076734601207821725

Level Eight

<http://www.lendingtree.com/stm/offers/realty.asp?realtysource=RC01LT01&partner=>

0.09989039857073354

<http://www.lendingtree.com/stm/default.asp>

0.02467731838074374

<http://www.lendingtree.com/stm/aboutlt/default.asp>

0.06231987177166482

<http://www.lendingtree.com/stm/joinus/default.asp>

0.08875499841153064

<http://www.lendingtree.com/stm/contactus/default.asp>

0.13887090923300616

<http://www.lendingtree.com/stm/aboutlt/investor/default.asp>

0.2135049458812595

D-2 “WWW Conference” Case (HITS)

Level One

<http://search.news.yahoo.com/search/news?p=conference+www+&c=news>

0.30751 61142110977

<http://www.sys-con.com/webwest/>

0.11375 166226648216

<http://search.news.yahoo.com/search/news?p=conference+www+&hc=1&hs=31>

0.089570527213745507

<http://www.iisg.nl/~w3vl/>

0.01277 8443675237054

<http://www.www2002.org/>

0.00996 7715504353978

<http://www oulu.fi/~spaceweb/lib/>

0.00704 1720594491317

Level Two

<http://www2002.org>

0.048551716263302314

<http://www.w3.org>

0.010579748743324756

<http://www.apple.com>

0.056204915198912761

<http://www.research.ibm.com>

0.023143200376022903

<http://www.htdc.org>

0.001322468592915594

<http://www.verity.com>

0.001674890334598886

Level Three

<http://www.hawaii.rr.com>

0.01142884550192782

<http://www.ptc.org>

0.03372538964230437

<http://www.w3c.it/>

0.0760115813871185

<http://validator.w3.org/>

0.031874045252493856

<http://www.keio.ac.jp/>

0.020974735624734116

<http://www.mac.com/>

0.009560973635930557

Level Four

<http://www.hawaii.rr.com/memberservices/default.htm>

0.04107749616816659

<http://www.hawaii.rr.com/memberservices/software.htm>

0.039692637757511605

<http://www.aroundhawaii.com/asp-bin/news/default.asp>

0.2538773774204416

http://www.oceanic.com/page_server/Oceanic/Road%20Runner/3F8E843C18DD8F64E5F491025C.html

0.04821985151277025

<http://www.hawaii.rr.com/rrhelp>

0.026371104067727594

<http://www2002.org/>

0.026972827521715318

Level Five

<http://www.hawaii.rr.com/RoadRunner/Status/default.htm>

0.05934491440632041

<http://www.hawaii.rr.com/rrhelp/>

0.014622024247747611

<http://www.hawaii.rr.com/rrhelp/rrclass.htm>

0.005120508634981714

<http://www.microsoft.com/security/>

0.03749260234656216

<http://www.webopedia.com>

0.09046231921801028

http://www.oceanic.com/page_server/Oceanic/Road%20Runner/Road%20Runner/155B7D351774EFC2E6A8AC449D.html

0.02481343588318306

Level Six

<http://www.aroundhawaii.com/default.asp>

0.024194426361456488

http://www.aroundhawaii.com/interact/contact_us.htm

0.031664478043792084

<http://www.aroundhawaii.com/streamingmedia/default.htm>

0.031567378201834466

<http://www.aroundhawaii.com/communities/default.htm>

0.055772098038282814

<http://www.aroundhawaii.com/interact/default.htm>

0.027672176891873297

<http://www.aroundhawaii.com/sports/default.htm>

0.08047013343168931

Level Seven

<http://www.hawaii.rr.com/memberservices/default.htm>

0.03299234261331837

<http://www.hawaii.rr.com/memberservices/software.htm>

0.030764656993643617

<http://www.aroundhawaii.com/asp-bin/news/default.asp>

0.16928134155875457

http://www.oceanic.com/page_server/Oceanic/Road%20Runner/3F8E843C18DD8F64E5F491025C.html

0.0294150427656851

<http://www.hawaii.rr.com/rrhelp>

0.019034730613269935

http://www.oceanic.com/page_server/Oceanic/Help%20Desk/Contact%20Help%20Desk/48569B224A0409E9E5159FCCC7.html
0.045242960989279664

Level Eight

<http://www.hawaii.rr.com/RoadRunner/Status/default.htm>

0.06015648377250679

<http://www.hawaii.rr.com/rrhelp/>

0.01482198724491123

<http://www.hawaii.rr.com/rrhelp/rrclass.htm>

0.005190533977321771

<http://www.microsoft.com/security/>

0.03895718590194636

<http://www.webopedia.com>

0.09169943359935129

http://www.oceanic.com/page_server/Oceanic/Road%20Runner/Road%20Runner/155B7D351774EFC2E6A8AC449D.html

0.025152771184852585