Chapter 2 Literature Review

2.1 Link Structure

The structure of the web is like a directed graph, which the web pages represent the nodes of the graph and the links between pages are edges. Every page has some number of forward links (outedges) and backlinks (inedges) (see Figure 2.1).This kind of structure is similar to the citation of academic papers, that an important paper would be citied many times. Thus a web page is recommended by another page linking to it. Therefore, the importance of a page depends on how many backlinks a page has.

(1) Backlink: page A and B are backlinks of page C.
(2) Forward link: page A has a forward link to page C.

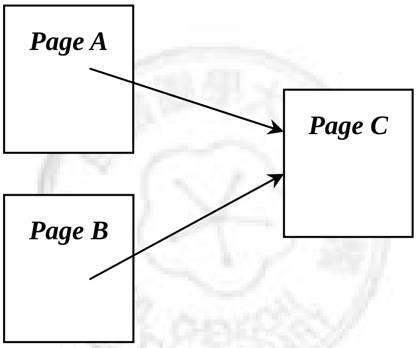

Figure 2.1: Link Structure
Source from Brin, S., & Page, L. (1998)

2.2 PageRank

PageRank is a well-known algorithm of Google using the link structure of the web to get the rank of the page. When a user submits a query to a search engine, the list result is ranked by the pagerank value of the page; the higher the pagerank, the more important the page.

If you just count the backlinks of the page as its rank value, it might be too simple. In order to make up for the drawback of in-links numbers only, PageRank takes account of the weighted value of links. Thus, PageRank has considered both the numbers of in-links and if the in-links are important.

2.2.1 PageRank algorithm

Brin, S., & Page, L. (1998) defined the algorithm of PageRank as below:

$PR(A) = (1-d) + d\ (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))$

where

PR(A) = the PageRank of page A

T1~Tn = all pages that link to page A

PR(Ti) = the PageRank of page Ti

C(Ti) = the numbers of pages which Ti links to

d = damping factor which can be set between 0 and 1。

PR(Ti)/C(Ti) = PageRank of Ti distributing to all pages that Ti links to

(1 - d) = to make up for the some pages that do not have any out-links to avoid loosing some PageRank.

2.2.2 Illustration of PageRank algorithm

PageRank is based on the link structure and is an iterative algorithm. A page would propagate its PageRank to the pages to which it links (See figure 2.2). There is a small problem with this simplified ranking function. Consider two web pages that point to each other but to no other page. And suppose there is some web page which points to one of them. Then, during iteration, this loop will accumulate rank but never distribute any rank (since there are no outedges). The loop forms a sort of trap which we call a rank sink (Lawrence, P., Sergey, B., Rajeev, M., & Terry, W. 1998). To make up for rank sink, add 1-d value to the algorithm.

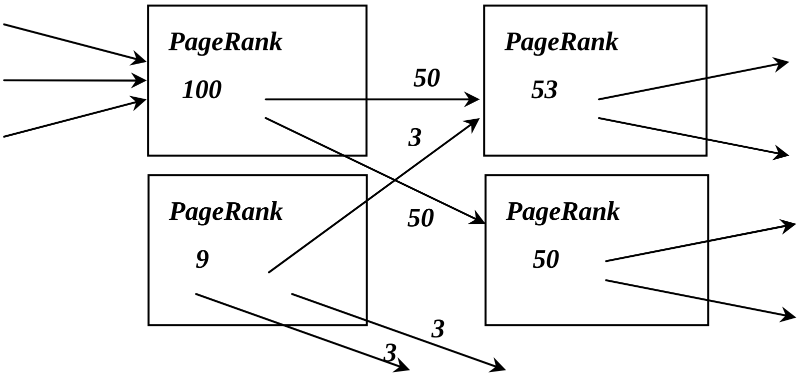

Figure 2.2: Illustration of PageRank algorithm
Source from Brin, S., & Page, L. (1998)

2.5.3 Example of PageRank

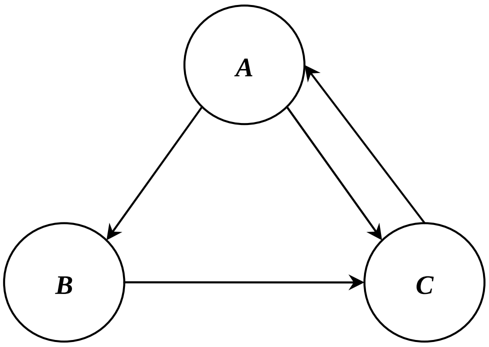

Figure 2.3: Example of PageRank

For instance, there is a small web containing page A, page B and page C, as shown in Figure 2.3. A links to B and B links to C. And C links to A. In order to compute PageRank, damping factor is set to 0.5. The computation is as follows.

PR(A) = 0.5 + 0.5 PR(C)

PR(B) = 0.5 + 0.5 (PR(A) / 2)

PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))

P(A), P(B) and P(C) are set to 1 initially and then iterated to get an approximate value as below, until the twelfth time, as below.

PR(A) = 14/13 = 1.07692308

PR(B) = 10/13 = 0.76923077

PR(C) = 15/13 = 1.15384615

Finally, we find out that the sum of PageRank of three pages is 3.

Table 2.1: PageRank of page A, B, C in each iteration

ith Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1.1484375
1	1	0.75	1.15283203
2	1.0625	0.765625	1.15365601
3	1.07421875	0.76855469	1.15381050
4	1.07641602	0.76910400	1.15383947
5	1.07682800	0.76920700	1.15384490
6	1.07690525	0.76922631	1.15384592
7	1.07691973	0.76922993	1.15384611
8	1.07692245	0.76923061	1.15384615
9	1.07692296	0.76923074	1.15384615
10	1.07692305	0.76923076	1.15384615
11	1.07692307	0.76923077	1.125

12	1.07692308	0.76923077	1.15384615

2.5.4 Generic Constructs of PageRank

After reviewing the PageRank algorithm, we can understand that the computation of the PageRank is based on the sum of the rank value of inlinks. We find out that the web page is a basic generic construct of the algorithm. In the PageRank algorithm, the hyperlink of the web page is another impact factor. The inlinks of the page are obtained by finding the hyperlinks of the web page so tag is an operation which operates the generic constructs “Web age”. Therefore, we summarize that “web page” is the generic construct and “tag” is the operation of the generic construct of the PageRank algorithm as shown in Table 2.2.

Table 2.2: Generic Constructs of PageRank

Generic Constructs	Web page
Operation of Generic Constructs	Tag

2.2 HITS ("hypertext induced topic selection")

Kleinberg, J. M. (1999), constructed the HITS algorithm based on link structure between pages to find authoritative pages, authorities, and hubs which link to many authorities. The detail of HITS will be described in the following section.

2.2.1 Features of HITS algorithm

(1) The model is based on links between web pages so that computing the authority weights of pages could result in the most authoritative pages.

(2) If a page has some authoritative in-link pages, it gets higher weighted value as well as being judged as a more important page.

(3) HITS is based on hubs and authorities. The authority and hub have mutually reinforcing relationships. A hub is a page that links to many authorities, so that an authority might be linked by many hubs. The relationship between hubs and authorities is shown in Figure 2.4.

(4) In order to lower cost and improve efficiency, HITS narrows the scope of pages that is the search result of a search engine and then constructs a subgraph of WWW by expanding the search result. Hopefully, we can get many authoritative pages.

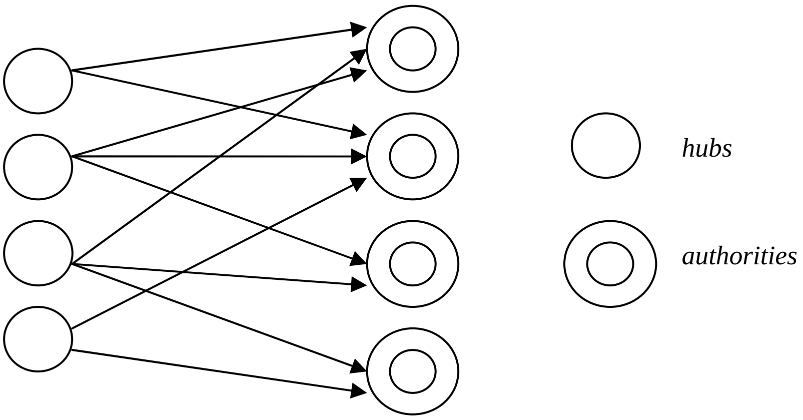

Figure 2.4: The Relationship Between Hubs and Authorities
Source:Kleinberg, J. M. (1999)

2.2.2 Scope of HITS

Kleinberg, J. M. (1999), the first step of HITS is to construct a focused subgraph of the WWW. Denote the collection V of hyperlinked pages as a directed graph G = (V;E) and the nodes correspond to the pages and a directed edge (p; q) = E, indicating the presence of a link from p to q. The out-degree of node p is the number of nodes to which it has links. The in-degree of p is the number of nodes that have links to it. From a graph G, we can isolate small regions, or subgraphs, in the following way. If W is a subset of the pages, we use G[W] to denote the graph induced on W: its nodes are the pages in W, and its edges correspond to all the links between pages in W. If we are searching on WWW now, and the query is as σ, we must determine the subgraph of the www so-called S_σ with the following properties.

(1) S_σ is relatively small.

(2) S_σ is rich in relevant pages.

(3) S_σ contains most (or many) of the strongest authorities.

Then, for a parameter *t* (typically set to about 200), we first collect the *t* highest-ranked pages for the query from a text-based search engine such as AltaVista or Hotbot. Refer to these *t* pages as the root set R_σ.We can increase the number of strong authorities in our subgraph by expanding R_σ to get the subgraph S_σ via the following steps.

(1) For each p of R_σ, add all pages *p* points to S_σ.

(2) For each p of R_σ, set a number d.

- If the number of all pages which link to p is less than d, add those to S_σ
- If the number of all pages which link to p is grater than d, add d pages linking to p to S_σ.

2.2.3 Mutually reinforcing relationships between hubs and authorities

There exits a mutually reinforcing relationship between hubs and authorities. A good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs.

2.2.4 The algorithm of HITS

(1) Authority weight of p is $x^{\langle p\rangle}$, and its hub weight is $y^{\langle p\rangle}$

(2) Through normalization, $\sum_{p\in S_\sigma}\left(x^{\langle p\rangle}\right)^2 = 1$, $\sum_{p\in S_\sigma}\left(y^{\langle p\rangle}\right)^2 = 1$.

(3) I Operation

$$x^{\langle p\rangle} \leftarrow \sum_{q:(q,p)\in E} y^{\langle q\rangle}$$

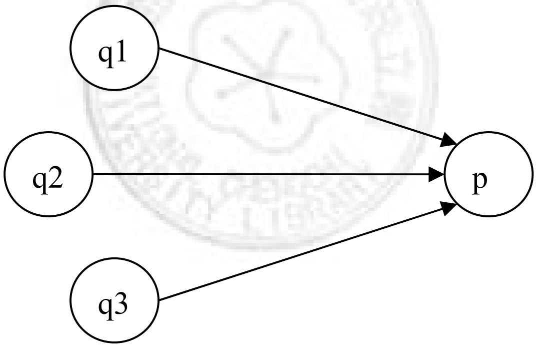

Figure 2.5: Authority weight p = hub weight q1 + hub weight q2+hub weight q3
Source: (Kleinberg, J. M. ,1999)

(4) O Operation

$$y^{\langle p\rangle} \leftarrow \sum_{q:(p,q)\in E} x^{\langle q\rangle}$$

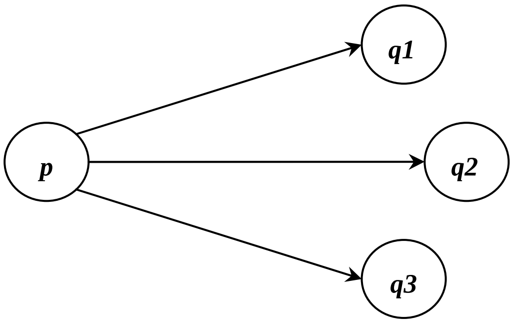

Figure 2.6: Hub weight p=authority weight q1+authority weight q2+authority weight q3
Source: (Kleinberg, J. M. ,1999)

(5) Iterative algorithm： The set weight of { $x^{\langle p\rangle}$ } is a vector x of G_σ, and the set weight of { $y^{\langle p\rangle}$ } is a vector y with the following iterative algorithm.

```
Iterate(G,k)
    G: a collection of n linked pages
    k: a natural number
Let z denote the vector (1, 1, 1,….., 1) $\in R_n$
Set $x_0 := z$:
Set $y_0 := z$:
For i = 1, 2,…… k
Apply the I operation to ($x_{i-1}, y_{i-1}$), obtaining new x-weights $x'_i$.
Apply the O operation to ($x'_{i-1}, y'_{i-1}$), obtaining new y-weights $y'_i$.
Normalize $x'_i$, obtaining $x_i$.
Normalize $y'_i$ , obtaining $y_i$.
End
Return ($x_k, y_k$).
```

(6) According to the iterative algorithm, the following procedure can be applied to filter out the top *c* authorities and top *c* hubs in the following simple way.

```
Filter(G,k,c)
    G: a collection of n linked pages
    k,c: natural numbers

($x_k, y_k$):=Iterate(G; k).
Report the pages with the c largest coordinates in $x_k$ as authorities.
Report the pages with the c largest coordinates in $y_k$ as hubs.
```

2.2.5 Generic Constructs of HITS

After reviewing the HITS algorithm, we can undertand that the computation of the authority is based on the sum of the hub value of inlinks. And the computation of the hub is based on the sum of the authority value of outlinks. We find out that web page is a basic generic construct of HITS. In the HITS algorithm, the hyperlink of the web page is also an impact factor. The inlinks and outlinks are obtained by finding the hyperlinks of the web page so tag is an operation which operates the generic construct “Web page”. Therefore, we summarize that “web page” is the generic construct and “tag” is the operation of the generic construct in the HITS algorithm as shown in Table 2.3.

Table 2.3: Generic Constructs of HITS

Generic Constructs	Web page
Operation of Generic Constructs	Tag

2.3 Improvement of HITS

2.3.1 The problems of HITS

Bharat identified two problems concerning HITS algorithm (Bharat,1998)：

(1) Mutually Reinforcing Relationships Between Hosts

Sometimes a set of documents on one host points to a single document on a second host. This drives up the hub scores of the documents on the first host and the authority score of the document on the second host. In the reverse case, where there is one document on a first host pointing to multiple documents on a second host, it creates the same problem.

(2) Top drift

We often find that the neighborhood graph contains documents not relevant to the query topic. If these nodes are well connected, the topic drift problem arises: the most highly ranked authorities and hubs tend not to be about the original topic.

2.3.2 BHITS

In order to improve two problems, Bharat improved the HITS algorithm as follows (Bharat, 1998).

(1) If there are k edges from documents on a first host to a single document on a second host, we give each edge an authority weight of $\frac{1}{k}$ (see Figure 2.7).

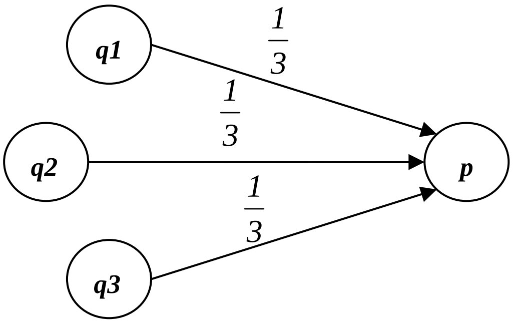

Figure 2.7: Authority weight of p = hub weight of q1 *1/3+hub weight of q2*1/3+hub weight of q3*1/3

Source: (Bharat, 1998)

(2) If there are l edges from a single document on a first host to a set of documents on a second host, we give each edge a hub weight of $\frac{1}{l}$ (see Figure 2.8).

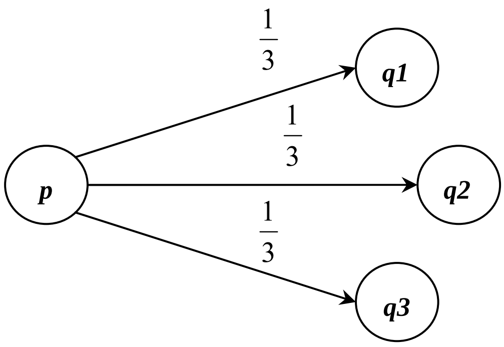

Figure 2.8: Hub weight of p =authority weight of q1*1/3+authority weight of q2*1/3+authority weight of q3*1/3

Source: (Bharat, 1998)

2.3.3 The algorithm of BHITS

Bharat (1998) defined the algorithm of BHITS as below:

(1) While the vectors *H* and A have not converged:

(2) For all n in N,

$$A[n] := \sum_{(n',n)\in N} H[n'] \times auth\_wt(n',n)$$

(3) For all *n* in N,

$$H[n] := \sum_{(n,n')\in N} A[n'] \times hub\_wt(n,n')$$

(4) Normalize the H and A vectors.

2.3.4 WBHITS (Weighted BHITS)

Longzhuang Li, Yi Shang, and Wei Zhang improved the HITS algorithm and BHITS algorithm as in the following description of the so-called WBHITS (Li, L., Shang, Y., & Zhang, W., 2002).

(1) For all $i' \in I$ which points to I,

$$a_i = \sum_{i'} w_{a_{i'}} \cdot h_{i'}$$

where

(i) If there is a root link whose in-degree is among the three smallest ones and whose out-degree is among the three largest ones, then set $w_{a_{i'}}$ to 4 for in-links of all the root links.

(ii) If there is a root link whose authority value is among the three smallest ones and whose hub value is among the three largest ones, set $w_{a_{i'}}$ to 4 for in-links of all the root links.

(iii) Otherwise, set all $w_{a_{i'}}$ to 1.

In the above two steps, usually the in-degree of a small-in-large-out link is as small as 0,1 and 2, while the out-degree can be more than several hundred.

(2) For all $i' \in I$ which is pointed by i,

$$h_i = \sum_{i'} w_{h_{i'}} \cdot a_{i'} \text{ ,}$$

where all the $w_{h_{i'}}$ values are set to 1.

2.3.5 Generic Constructs of BHITS and WBHITS

In the prior section, we have already summarized generic constructs of HITS. After reviewing the BHITS and the WBHITS algorithm, we can understand that the computation of the BHITS and the WBHITS is similar to the HITS algorithm. The difference between HITS and weighted based HITS is the variable “weight value”. Similar to HITS, we can conclude that the web page is the generic construct of BHITS and WBHITS. In addition, the hyperlink of the web page in BHITS and WBHITS is also an impact factor so the tag of the web page is the operation which operates the generic construct in the algorithm. Therefore, we summarize that “web page” is the generic construct and the “tag” is the operation of the generic construct of BHITS and WBHITS as shown in Table 2.4.

Table 2.4: Generic Constructs of BHITS and WBHITS

Generic Constructs	Web page
Operation of Generic Constructs	Tag

2.3.6 Combining the HITS-based algorithms with relevance scoring methods

Longzhuang Li, Yi Shang, and Wei Zhang (2002), combined the HITS-based algorithms with relevance-scoring methods. If s_i is the relevance score of a Web page i and h_i the hub value,

$s_i * h_i$ instead of h_i is used to compute the authority value of the Web pages to which it points. Similarly, if a_i is its authority value, $s_i * a_i$ instead of a_i is used to compute the hub values of Web pages that point to it. The relevance score, s_i, which is the combined four relevance scoring methods: VSM, TLS, Okapi and CDR introduced in the following sections.

2.3.6.1 Vector Space Model (VSM)

For a fixed collection of documents, an m-dimensional vector is generated for each document and each query from sets of terms with associated weights, where m is the number of unique terms in the document collection. Then, a vector similarity function, such as the inner product, can be used to compute the similarity between a document and a query.

The similarity $sim_{vs}(q, x_i)$, between query q and document x_i, can be defined as the inner product of the query vector Q and the document vector X_i:

$$sim_{vs}(q, x_i) = Q \cdot X_i = \frac{\sum_{j=1}^{m} v_j \cdot w_{ij}}{\sqrt{\sum_{j=1}^{m} (v_j)^2 \cdot \sum_{j=1}^{m} (w_{ij})^2}}$$

where

N = the total number of documents in the collection

y_j = each term of the query

x_i = each document

f_{ij} = the occurrence of term y_j in the documents x_i

d_j = the number of documents containing term y_j

$$v_j = \begin{cases} \log\left(\frac{N}{d_j}\right) & y_j \text{ is a term in q} \\ 0 & otherwise \end{cases}$$

$$w_{ij} = f_{ij} \cdot log\left(\frac{N}{d_j}\right)$$

$$g_j = log\left(\frac{N}{d_j}\right)$$, Inverse document frequency

- Constraints ：The VSM method cannot be applied directly in evaluating the precision of search engines because it can not determine N 和d_j.

2.3.6.2 Generic Constructs of HITS based-VSM

Similar to HITS, the generic constructs of the HITS based algorithm-VSM includes the web page and tag. In addition, after reviewing the VSM algorithm, we can understand that scoring of VSM is based on term frequency. Term is the input of web search so term frequency is the crucial factor to the search algorithm to get the relevant page. Therefore, the term is a generic construct of VSM. The “term frequency” is obtained by counting the number of the term occurring in the page so “count” is an operation which operates the generic constructs in the algorithm. Therefore, we summarize that “web page” and “term” are the generic constructs of the HITS based-VSM. And “tag” and “count” are the operations of the generic constructs of HITS based-VSM as shown in Table 2.5.

Table 2.5: Generic Constructs of HITS based-VSM

Generic Constructs	Web page, Term
Operation of Generic Constructs	Tag, Count

2.3.6.3 Okapi Similarity Measurement (Okapi)

(1) The algorithm

$$sim_o(q,x_i)=Q\cdot X_i=\sum_{j=1}^{m}v_j\cdot w_{ij}$$

$$w_{ij}=\frac{f_{ij}\cdot log\left(\frac{N-d_j+0.5}{d_j+0.5}\right)}{2\cdot\left(0.25+0.75\cdot\frac{dl}{avdl}\right)+f_{ij}}$$

where

dl = the length of the document (in bytes)

$avdl$ = the average document length in the collection (in bytes)

(2) Constraints: Similarly to the VSM method, Okapi similarity measurement cannot be applied directly in evaluating the precision of the search engines. In addition, the average length of a Web document (avdl) is estimated as to be 10,939 bytes.

2.3.6.4 Generic Constructs of HITS based-Okapi

Similar to the HITS, the generic constructs of HITS based -Okapi includes the web page and tag. In addition, after reviewing the Okapi algorithm, we can understand that scoring of Okapi is based on term frequency and document length. Term is the input of web search so term frequency is the crucial factor to the search algorithm to find the relevant page. Document length is a variable of the algorithm. Therefore, we can conclude that the “term” is a generic construct of Okapi. The “term frequency” is obtained by counting the number of the term occurring in the page so “count” is an operation which operates the generic constructs in the algorithm. We summarize that “web page” and “term” are generic constructs of HITS based-Okapi. And “tag” and “count” are two operations of generic constructs of HITS based-Okapi as shown in Table 2.6.

Table 2.6: Generic Constructs of HITS based-Okapi

Generic Constructs	Web page, Term
Operation of Generic Constructs	Tag, Count

2.3.6.5 Cover Density Ranking (CDR)

In CDR, the results of phrase queries are ranked in the following two steps:

(1) Documents containing one or more query terms are ranked by coordination level, i.e., a document with a larger number of distinct query terms ranks higher. The documents are thus sorted into groups according to the number of distinct query terms each contains, with the initial ranking given to each document based on the group in which it appears.

(2) The documents at each coordination level are ranked to produce the overall ranking. The score of the cover set $\omega = \{(p_1, q_1), (p_2, q_2), \dots, (p_n, q_n)\}$ is calculated as follows:

$$S(w) = \sum_{j=1}^{n} I(p_j, q_j) \quad \text{and} \quad I(p_j, q_j) = \begin{cases} \dfrac{\lambda}{q_j - p_j + 1} & if \quad q_j - p_j + 1 > \lambda \\ 1 & otherwise \end{cases}$$

where

$p_j =$ the position of one term on a document

$q_j =$ the position of another term on a document, $q_j > p_j$

$(p_j, q_j) =$ an ordered pair over a document, called cover

λ = is a constant

2.3.6.6 Generic Constructs of HITS based-CDR

Similar to the HITS, the generic constructs of the HITS based algorithm-CDR includes the web page and tag. In addition, after reviewing the CDR algorithm, we can understand that scoring of Okapi is based on the position of the term. Term is the input of web search so the position of the term is the crucial factor to CDR algorithm to find the relevant page. Therefore, the “term” is a generic construct of CDR. The position of the terms is obtained by positioning the term in the page so “position” is an operation of generic constructs. We summarize that “web page” and “term” are generic constructs of the HITS based-CDR. And “tag” and “position” are two operations of generic constructs of HITS based-CDR as shown in Table 2.7.

Table 2.7: Generic Constructs of HITS based-CDR

Generic Constructs	Web page, Term
Operation of Generic Constructs	Tag, Position

2.3.6.7 Three-Level Scoring Method (TLS)

The TLS method computes the relevance of a Web page to a query in the following two steps:

(1) Given a query phrase q with n terms and a Web page x, a raw score is calculated as $A(q,x)$:

$$A(q,x) = \frac{t_n \cdot k^{n-1} + t_{n-1} \cdot k^{n-2} + ... + t_1}{k^{n-1}}$$

where

k = a constant, corresponding to the weight for longer phrases;

$t_i, 1 < i < n$:the number of occurrence of the sub-phrases of length i, i.e., containing i terms.

(2) Convert the raw score $A(q,x)$ to a three-level relevance score through thresholding, with value 2 for relevant, 1 for partially relevant, and 0 for irrelevant:

$$sim_{tls}(q,x) = \begin{cases} 2 & if \quad A(q,x) \geq \Theta \\ 1 & if \quad \Theta > A(q,x) \geq \alpha\Theta \\ 0 & if \quad A(q,x) < \alpha\Theta \end{cases}$$

where $0 < \alpha < 1$ and Θ is a constant

2.3.6.6 Generic Constructs of HITS based-TLS

Similar to the HITS, the generic constructs of the HITS based algorithm-TLS includes the web page and the tag. In addition, after reviewing the TLS algorithm, we can understand that scoring of TLS is based on the sub-phrase frequency. Term is the input of web search and phrase is the combination of terms so the sub-phrase frequency is the crucial factor to the TLS algorithm. Therefore, we can conclude that the “phrase” and the “term” are two generic constructs of the TLS. The sub-phrase frequency is obtained by counting the number of the sub-phrase occurring in the page so “count” is an operation of generic constructs. We summarize that “web page”, “term” and “phrase” are the generic constructs of the HITS based-TLS. And “tag” and “count” are the operations of generic constructs of HITS based-TLS as shown in Table 2.8.

Table 2.8: Generic Constructs of HITS based-TLS

Generic Constructs	Web page, Term, Phrase
Operation of Generic Constructs	Tag, Count

2.4 TREC Web Track

The Text Retrieval Conference (TREC), started in 1992 as part of the TIPSTER Text program, is to support research within the information retrieval community by providing the infrastructure (test collections, evaluation methodology, etc.) necessary for large-scale evaluation of text retrieval methodologies. A TREC workshop consists of a set tracks, areas of focus in which particular retrieval tasks are defined. Web Track is a track featuring search tasks on a document set that is a snapshot of the World Wide Web. This Web track last ran in TREC 2004. The Web Track of 2004 and 2004 will be described in the following sections:

2.4.1 The Web Track of TREC 2003

The TREC 2003 web track consisted of both a non-interactive stream and an interactive stream. Both streams worked with the .GOV test collection. The non-interactive stream continued an investigation into the importance of homepages in Web ranking, via both a Topic Distillation task and a Navigational task. In the topic distillation task, systems were expected to return a list of the homepages of sites relevant to each of a series of broad queries. This differs from previous homepage experiments in that queries may have multiple correct answers. The navigational task required systems to return a particular desired web page as early as possible in the ranking in response to queries. In half of the queries, the target answer was the homepage of a site and the query was derived from the name of the site (Homepage finding) while in the other half, the target answers were not homepages and the queries were derived from the name of the page (Named page finding). The two types of query were arbitrarily mixed and not identified. The interactive stream focused on human participation in a topic distillation task over the .GOV collection. Studies conducted by the two participating groups compared a search engine using automatic topic distillation features with the same engine with those features disabled in order to determine whether the automatic topic distillation features assisted the users in the performance of their tasks and whether humans could achieve better results than the automatic system.

At the Web track of the TREC 2003, Microsoft Research Asia (MSRA) report their system and methods on the topic distillation task and the home/named page finding task. MSRA designed a Web search platform to conduct their experiments. They proposed a novel block-based HITS algorithm -“Block-based HITS” to solve the noisy link and topic drifting problems of the classical HITS algorithm. The basic idea is to segment each Web page into multiple semantic blocks using a vision-based page segmentation algorithm they developed before. And they constructed a hierarchical site map for each website compression technique to select the relationships of Web pages in the .GOV dataset. They apply a site compression technique to select the most suitable entry pages for websites among the retrieval results and return these entry pages as top-ranked page. MSRA use Okapi’s BM2500 as their fundamental relevance ranking function and made some important modifications and augmentations to set different weight to different term types and formats. For example, they use term proximity to adjust the relevance scores.

2.4.2 The Web Track of TREC 2004

The tasks of TREC-2003 involved queries of three types: “Topic distillation”(TD),” Homepage finding”(HP) and “Named page finding”(NP). The main experiment in TREC 2004 involved processing a mixed query stream, with an even mix of each query type studied in TREC-2003: 75 homepage finding queries, 75 named page finding queries and 75 topic distillation queries. The goal was to find ranking approaches which work well over the 225 queries, without access to query type labels.

Microsoft Research Asia (MSRA) reports their experiments on the mixed query of Web Track at TREC 2004. They test a set of new technologies to test some new features of Web pages to see if they are useful to retrieval performance. Title extraction, sitemap based feature propagation, and URL scoring are of this kind. Another effort is to propose a new function or an algorithm to improve relevance or importance ranking. They found that a new link analysis algorithm name HostRank that can outweigh PageRank for topic distillation queries based on their experimental results. Eventually, the linear combination of multiple scores with normalizations is tried to achieve stable performance improvement with mixed queries.

2.5 Summary

In this section, we summarize that the algorithms we have reviewed in the chapter two (See Table 2.9). The Collection of generic constructs and operations of generic constructs of the algorithms are summarized in Table 2.10 and Table 2.11. In addition, we analyze that the collection of generic constructs and operations of generic constructs of the algorithms of the MSRA at Web Track of TREC 2003 and 2004 as shown in Table 2.12 and Table 2.13.

Table 2.9: Summary of Algorithms

Algorithm	Objective	Methodology	Advantage	Disadvantage	Application
PageRank	Finding the most authoritative page	Link structure analysis	Not just counting term frequency.	Easy to be manipulated by a site host.	Personalized search engine.
HITS	Finding authorities and hubs. ● Authorities are those containing rich relevant information and recommended by many pages. ● Hubs are those pages that link to many related authorities.	Link structure analysis	Not just focusing on finding authority page.	● Mutually Reinforcing Relationships Between Hosts：Drive up the hub value and authority value. ● topic drift：small-in-large-out pages would include many irrelevant pages.	Broad-Topic queries.
BHITS	Improved Kleinberg's HITS algorithm by giving a document an authority weight.	Link structure analysis	Resolve the Mutually Reinforcing Relationships Between Hosts。	topic drift：Small-in-large-out pages would include many irrelevant pages.	Broad-Topic queries
WBHITS	Add more weights to the in-links of root set links if a small-in-large-out link exists.	Link structure analysis	Prevent topic drift caused by small-in-large-out pages。		

HITS based —VSM	Improved HITS algorithm by adding the relevance score between pages and query term so if the page is relevant, it gets high relevance score. Otherwise, it gets the low score.	● Link structure analysis ● Content relevance scoring	Not only considers term frequency but gets hubs and authorities relevant to query topic.	VSM method cannot be applied directly in evaluating the precision of search engines.	Finding pages which just have one term of the query phrase.
HITS based — Okapi	Improved HITS algorithm by adding the relevance score between pages and query term so if he page is relevant, it gets high relevance score. Otherwise, it gets the low score.	● Link structure analysis ● Content relevance scoring	Not only considers both term frequency and the length of the document but also gets authorities and hubs relevant to query topic.	Okapi method cannot be applied directly in evaluating the precision of search engines.	Finding pages which just have one term of query phrase.
HITS based — CDR	Improved HITS algorithm by adding the relevance score between pages and query term so if he page is relevant, it gets high relevance score. Otherwise, it gets the low score.	● Link structure analysis ● Content relevance scoring	Considers both the number of terms pages have and the position of terms in the document so it can find the most relevant authorities and hubs.	CDR method cannot be applied directly in evaluating the precision of search engines.	Finding pages which just have each term of query phrase.
HITS based algorithm — TLS	Improved HITS algorithm by adding the relevance score between pages and query term so if he page is relevant, it gets high relevance score. Otherwise, it gets the low score.	● Link structure analysis ● Content relevance scoring	Not only considers term frequency but also the position and sequence of terms in the document.		Finding the exactly matching page.

Table 2.10: Generic Constructs of Algorithms

Generic Constructs	PageRank	HITS	BHITS	WBHITS	HITS based －VSM	HITS based －Okapi	HITS based － CDR	HITS based －TLS
Web page	√	√	√	√	√	√	√	√
Term					√	√	√	√
Phrase								√

Table 2.11: Operations of Generic Constructs of Algorithms

Operations of Generic Constructs	PageRank	HITS	BHITS	WBHITS	HITS based －VSM	HITS based －Okapi	HITS based － CDR	HITS based － TLS
Count					√	√		√
Position							√	
Tag	√	√	√	√	√	√	√	√

Table 2.12: Generic Constructs of MSRA at Web Track of TREC 2003 and 2004

Generic Construct	Microsoft Research Asia At Web Track of TREC 2003	Microsoft Research Asia At Web Track of TREC 2004			
	Block-based HITS	Title extraction	Sitemap based feature propagation	URL scoring	HostRank
Web page	√	√	√	√	√
Term					
Phrase					

Table 2.13: Operations of Generic Constructs of MSRA at Web Track of TREC 2003 and 2004

Generic Construct	Microsoft Research Asia At Web Track of TREC 2003	Microsoft Research Asia At Web Track of TREC 2004			
	Block-based HITS	Title extraction	Sitemap based feature propagation	URL scoring	HostRank
Count					
Position		√			
Tag	√	√	√	√	√