하이퍼링크 구조를 이용한 웹 검색의 순위 알고리즘에 관한 연구

The Study on the Ranking Algorithm of Web-based Searching Using Hyperlink Structure

김 성 희* · 오 건 택**
Sung-Hee Kim · Gun-Teak O

차 례

1. 서 론
2. 이론적 배경
3. PageRank 알고리즘 및 HITS 알고리즘 분석
4. PageRank 알고리즘 및 HITS 알고리즘
5. 결 론
• 참고문헌

초 록

본 연구에서는 하이퍼 링크 구조를 이용한 웹 검색 알고리즘에 대해 살펴 본 후 페이지 품질을 측정하기 위해 웹의 하이퍼 구조를 이용하고 있는 알고리즘인 HITS와 PageRank를 분석하였다. 이어서 이들 방법을 이용한 검색 엔진인 Google과 Ask.com을 검색 알고리즘의 특성을 기준으로 분석하였다. 이런 연구는 미래의 웹 문서의 중요도를 평가하는 데 기초자료로 활용할 수 있으며, 웹 정보검색의 검색성능을 향상시키는 시스템 개발에 도움이 될 수 있을 것이라 생각한다.

키 워 드

웹 검색, 순위, 검색 엔진, 페이지랭크, HITS, 구글, 애스크닷컴

* 중앙대학교 문헌정보학과 부교수
(Associate professor, ChungAng University, Seonghee@cau.ac.kr)
** 한국과학기술연구원(KIST) 경영기획실장
(Management Planning Manager, KIST, ktoh@kist.re.kr)
• 논문접수일자 : 2006년 5월 31일
• 게재확정일자 : 2006년 6월 17일

ABSTRACT

In this paper, after reviewing hyperlink based ranking methods, we saw various other parameters that effect ranking. Then, We analyzed the PageRank and HITS(Hypertext Induced Topic Search) algorithm, which are two popular methods that use eigenvector computations to rank results in terms of their characteristics. Finally, google and Ask.com search engines were examined as examples for applying those methods. The results showed that use of Hyperlink structure can be useful for efficiency of web site search.

KEYWORDS

Web search, Ranking Algorithm, Pagerank, HITS, Google, Ask.com

1. 서 론

인터넷의 급속한 성장은 시 · 공간의 제약을 극복할 수 있는 정보가상공간을 만들게 되었으며, 이러한 환경은 정보의 집합장소라고 대별될 수 있는 웹이라는 대규모 정보공간을 이루었다. 그러나 방대한 정보량은 선별의 어려움이라는 역기능을 창출하기도 하였으며, 이러한 정보의 역기능으로는 사용자에게 선택이라는 혼란을 발생시켰다.

인터넷상에서 존재하는 문서의 수는 기하급수적으로 증가하고 있으며, 그 수를 정확히 알기는 힘들어지고 있다. 끝없이 쌓여가고 생성되는 문서들 속에서 자신이 원하는 정보를 찾는 것은 점점 더 중요한 일이 되고 있다. 이러한 정보의 홍수 속에서 원하는 정보를 얻기 위해서 다양한 검색 엔진이 사용되고 있지만, 검색결과 역시 상당한 양이어서 이용자는 다시 여기서 중요한 페이지를 찾기 위해 많은 시간과 노력을 들여야만 한다. 사용자가 자신에게 필요한 정보를 쉽고, 빠르게 찾기 위해 많은 연구가 수행되어 왔는데, 일반적으로 대부분의 검색 엔진의 시스템들은 웹 페이지의 검색 및 순위결정에 내용기반(content based) 방법을 사용하고 있다. 내용기반검색방법은 문서 내의 모든 단어에 대해 단어의 빈도수와 출현위치를 이용하는 것이다. 웹 문서인 HTML의 경우, 단어뿐만 아니라 하이퍼링크, 이미지, 사운드 등 다른 요소들이 많이 포함되어 있다. 따라서 단순히 사용자가 던진 질의어를 한두 개 더 많이 포함하고 있는 페이지가 좋은 페이지가 아니라, 다수의 사용자에 의해 검증된 대표성이 높거나 인기도가 높은 페이지가 사용자가 더 선호하는 페이지이다. 예를 들어, 한국인 사용자라면 대개의 경우 "야후"라는 단어로 검색을 하는 경우에는 'kr.yahoo.com' 이 최상위에

출력되는 것을 원할 것이고, "옥션"이라는 단어로 검색을 하는 경우에는 'www.auction.co.kr'이 최상위에 출력되는 것을 원할 것이다. 그러나 현재 웹에 존재하는 문서 중에서 '야후', '옥션' 등의 단어를 포함한 문서는 헤아릴 수 없을 정도로 많으며, 이들 검색어에 해당하는 웹 문서 검색 엔진의 검색결과 역시 엄청난 숫자이다.

이러한 내용기반 방식 순위방법의 한계를 극복하기 위하여 새로운 검색기법 및 순위결정 방법을 도입한 이른 바 '제 2세대 검색 엔진'이라 불리는 새로운 검색 시스템에 대한 연구가 활발하게 진행되고 있다. 이러한 시스템들은 하이퍼링크 구조를 분석하거나, 사용자의 행위를 관찰하여 얻어진 정보를 이용함으로써 기존의 검색 엔진과는 전혀 다른 순위결정 개념을 사용하여 주목을 받고 있다. 이러한 연구에 힘입어 차세대 검색 엔진인 구글(Google), 애스크닷컴(Ask.com)은 사용자들에게 많은 호응을 얻고 있다. 차세대 검색 엔진들은 전통적 방식인 내용기반 방식의 문제점들을 해결하여, 사용자의 질의에 대해 적합한 웹 페이지를 추출하는 방식을 채택하고 있다. 따라서 본 연구에서는 하이퍼링크 구조를 이용한 웹 검색 알고리즘 유형에 대해 살펴보고, 유형별로 대표되는 검색 엔진의 순위 알고리즘을 분석하고자 한다. 이런 연구는 미래 웹 문서의 중요도를 평가하는 데 기초자료로 활용할 수 있으며, 웹 정보검색의 검색성능을 향상시키는 시스템 개발에 도움이 될 수 있을 것이라 생각한다.

2. 이론적 배경

2.1 검색순위 알고리즘

하이퍼링크 환경에서 문서의 링크 정보에 대한 연구는 크게 전역 링크 정보연구와 지역 링크 정보연구로 나눌 수 있다. 전역 링크 정보연구는 현재 Google 검색 시스템에서 사용하고 있는 PageRank 알고리즘과 같이 전역의 링크 정보를 이용하여 전체문서에 가중치를 부여하는 연구이고, 지역 링크 정보연구는 Kleinberg의 HITS 알고리즘처럼 사용자의 초기질의에 대한 기본 검색자료를 이용하여 선정된 기본 문서들과 관련된 지역적 링크 정보를 이용하는 연구가 있다.

2.1.1 PageRank 알고리즘

PageRank는 링크를 통해서 문서의 순위가 전파되어 나가는 개념이다. T1, ..., Tn이 페이지 A를 가리키고 있을 때, 페이지 A는 다음 수식과 같은 페이지 순위(PR)를 가진다.

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

위 수식에서 d는 사용자가 특정 페이지에서 만족하지 못하고, 다른 페이지로 이동할 확률을 말하며, $C(T_1)$는 T_1이 가리키는 문서 개수를 나타낸다. Google에서는 사용자가 질의어로 입력하는 단어를 모두 가지는 문서에 대해서

미리 계산해 놓은 PageRank를 가지고 순위를 결정한다. 즉, 색인어 정보에서는 동일하다고 생각되는 문서에 대해서 링크 정보를 이용해서 우선순위를 결정하는 방식이다(Brin and Page 1998).

2.1.2 HITS 알고리즘

HITS 알고리즘은 링크를 분석하여 웹 문서에 대한 authority와 hubs로서의 두 종류의 속성 값을 계산한다. Authorities는 제시된 주제에 대해 중요한 정보를 갖고 있는 문서라는 것을 의미하며, hubs는 authorities들로 연결하는 많은 수의 링크들을 가진 문서라는 것을 의미한다. 이를 위해 웹 페이지들 상호간에 연결된 링크의 빈도수로부터 authorities를 결정하며, 역으로 hubs를 알 수 있다.

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

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

〈수식 1〉

먼저 질의로부터 의미 기반의 검색 엔진에서 얻어진 검색결과의 웹 문서 집합 N을 HITS의 입력으로 사용한다. 위 〈수식 1〉에서 A[n]는 $n \in N$인 웹 문서 n에 대한 authorities score이고, H[n]는 hub score이다. 위의 연산을 반복적으로 수행하면, A[n]와 H[n]는 최종적인 authority score와 hub score로 수렴하게 되며, 이는 Kleinberg에 의해 증명되었다(Kleinberg 1998).

2.1.3 명성평가방법

명성평가기법은 주제에 관한 웹 페이지의 명성을 평가하는 방법으로, 웹 페이지가 얼마나 명성이 있는가를 계산하는 방법이다(Agichtein, E. S, Lawrence, and L. Gravano 2001).

명성평가기법은 검색 질의에 대한 페이지의 순위를 측정하기 위해 penetration과 focus라는 두 가지 비율을 사용한다. 주제를 t라 하고 페이지를 p라고 할 때, 주제 t에 대한 페이지 p 의 penetration은 페이지 p를 가리키며, 주제 t 인 페이지 수를 주제 t에 관한 전체 페이지의 수로 나눔으로써 측정된다. 주제 t에 관한 페이지 p의 focus는 페이지 p를 가리키며, 주제 t인 페이지 수를 페이지 p를 가리키는 페이지의 수로 나누어 측정된다. penetration은 주제 t인 임의의 페이지가 페이지 p를 가리키는 임의의 페이지가 주제 t일 확률을 말한다.

2.2 검색순위에 영향을 미치는 요소

인터넷 사용자들이 자신이 원하는 정보를 찾는 가장 일반적인 방법으로 검색 엔진을 가장 많이 사용하고 있으나, 질의어에 대한 검색결과는 수백에서 수백만건에 이르고 있다. 사용자 측면에서는 무수한 검색결과 중에서 자신이 찾고자 하는 정보와 무관한 내용이 나올 수도 있다. 따라서 검색 엔진들은 보다 정확한 검색결과를 보여주기 위하여 질의어의 위치와 연

관성 및 상관성 등이 있는 웹 페이지가 상위에 위치하도록 독특한 순위결정방법을 적용하고 있다(Schwartz and Candy 1998).

검색 엔진에서 높은 순위에 위치하기 위해서는 해당 사이트를 검색 엔진의 순위기법에 맞도록 최적화해야 한다. 사이트를 최적화하기 위해서 가장 중요한 것은 검색 엔진 사용자가 어떤 키워드나 구로 검색을 하였으며, 어떤 검색 엔진을 사용했는 지를 반드시 파악해야 한다. 이 방법은 서버의 로그 파일을 보거나 각종 통계 프로그램을 이용하여 분석할 수 있다.

Yahoo와 같은 디렉터리 검색 엔진의 경우 등록된 사이트들은 일정한 순서에 의해 리스트되어 있으며, 이러한 순위기법은 로봇 검색 엔진에 비해 상당히 간단하다. Yahoo의 경우 기호, 숫자, 알파벳의 순서대로 사이트들이 배열되어 있기 때문에 자신의 주제와 가장 부합된 카테고리 순서배열에 맞게 사이트 등록내용을 기입하면 상위순위에 위치할 수 있다. 하지만 디렉터리 검색 엔진의 경우 사이트 등록이 걸리는 시간이 상당히 길며, 등록이 쉽게 이루어지지 않는 경우가 많다. 일반적인 로봇 검색 엔진은 특정 URL을 순회하여 정보를 수집하고, 수집된 정보를 바탕으로 이를 순위기법에 적용하여 사용자에게 검색결과를 제공한다. 특정 질의어에 대한 검색결과 페이지는 일반적으로 한 페이지에 10개씩 나누어서 보여준다. 대부분의 검색자는 검색결과의 상위 30개 이내의 사이트만을 방문하고 있다(Filman 1998). 따라서 순위가 뒤로 갈수록 검색자가 방문할 가능성은 낮아진다. 상위순위를 차지하기 위해서는 각 검색 엔진별 순위기법을 정확히 파악하고 있어야 하는데, 검색 엔진의 순위기법은 검색 엔진마다 각기 조금씩 차이가 있지만, 대부분의 로봇 검색 엔진에서 사용되는 순위기법은 공통적인 요소가 많다. 다음은 검색결과 순위에 영향을 미치는 요소들이다.

(1) 메타 태그: 많은 검색 엔진들이 메타 태그를 순위기법에 적용한다. 알타비스타가 대표적인 이 요소를 적용하는 엔진이라 할 수 있다. 따라서 사이트를 검색결과에 우선출력하려면 메타 태그와 타이틀을 해당 사이트에 가장 적절한 키워드와 구로 작성하여야만 높은 순위에 위치할 수 있다. 최근에 검색 엔진들은 웹 브라우저에 보이는 문구나 내용도 순위기법에 적용하고 있다.

(2) 단어의 출현 위치와 빈도수: 얼마나 자주 검색 키워드가 사이트의 제목과 내용에 나오는가에 따라 순위가 결정된다. 자신의 사이트와 연관된 키워드를 메타 태그나 HTML 문서 내에 특정 키워드가 많을 경우 검색결과의 상위에 위치하게 될 가능성이 높게 된다. 일부 사이트들은 키워드의 빈도수를 높이기 위해 HTML 문서 내에 키워드를 반복하여 여러 번 입력하는 방법을 사용하는데, 과거에는 이런 방법으로 키워드 빈도를 높일 수 있었으며, 이를 위해 키워드 글자의 색깔을 배경색과 같게 하여 사람의 눈에는 보이지 않게 하면서(invisible text)

검색 엔진을 속이는 경우도 있었다. 그러나 최근 검색 엔진들은 HTML 문서 내에 키워드가 일정 수 이상 나타나 스팸(spam)의 성격이 있다고 판단되면 해당 사이트의 우선순위를 내리거나, 심지어 robot이 index 대상에서 제외시켜 버리는 경우도 있다. 대부분의 검색 엔진들이 홈페이지에 검색되는 전체 키워드를 비교하여 가장 중심이 되는 키워드에 순위를 매겨 검색결과를 나타내기 때문에 HTML 문서 내에 다양한 문장들을 구성하여 핵심이 되는 키워드를 적절히 문구에 삽입시키면 높은 순위를 얻을 수 있다. 키워드 출현순서는 얼마나 일찍 검색 키워드가 사이트의 제목과 내용 중에서 나타나는가에 따른 검색결과로 일반적으로 HTML 문서 내에 제목이나 내용의 첫 번째 줄에 키워드를 기입하면 높은 순위의 검색결과를 얻을 수 있다. 현재 대부분의 검색 엔진 순위는 질의어에 연관된 페이지의 단어 출현위치와 빈도수를 이용한다. 이러한 키워드의 위치와 빈도의 적용여부는 각 검색 엔진마다 약간씩의 차이점은 있고, 어느 정도의 가산점이 부여되기는 하지만, 다른 요인들에 의해서도 순위가 결정된다. 이와 같이 대부분의 검색 엔진 사이트의 타이틀과 사이트를 구성한 키워드에 따라 점수를 산정하여 검색순위를 산정하기 때문에 상위에 위치하기 위해서는 키워드를 효과적으로 배치시켜야 한다. 일부 검색 엔진에서는 이미지 부분의 태그가 검색 질의어와 상관성이 있으며, 순위에 가산점을 주어 반영하고 있고, 검색 엔진의 디렉터리에 등록된 사이트를 그렇지 않은 사이트보다 높은 점수를 주는 검색 엔진도 있다.

(3) 메타 리프레시(meta refresh): 사이트의 구성을 특정 검색 엔진의 상위순위 진입을 위해 구성된 페이지는 도어웨이 페이지(doorway page) 혹은 브리지 페이지(bridge page)라고 불린다. 도어웨이 페이지의 특징은 몇 초 안에 자동으로 사이트의 초기화면 등으로 이동하는 메타 리프레시(meta refresh)를 사용하고 있다. 일부 검색 엔진에서는 스팸을 방지하기 위해서 메타 리프레시 페이지가 있는 부분은 인덱싱하지 않고 있으나, Google과 같은 링크 인기도에 기반을 둔 순위 시스템을 적용하고 있는 검색 엔진은 메타 리프레시에 제한 스팸 규정을 두지 않고 있다.

(4) 링크 구조: 거의 대부분의 검색 엔진에서는 다른 페이지에 얼마나 링크가 되어있는 지 분석을 하여 해당 사이트의 인기도를 결정할 수 있으며, Google과 같은 검색 엔진에의 순위기법에서 상당히 중요한 요인으로 작용하고 있다.

2.3 선행연구

웹 문서에서 링크는 웹 문서 간을 연결하는 역할을 수행하므로 웹 관련연구에 있어서 링크는 매우 중요한 요소이다. 링크 정보는 다양하

계 이용될 수 있는데, Wang과 Kitsuregawa(2001)는 링크를 이용한 클러스터링을 제안하였다. 이 연구에서는 웹 문서 p와 q가 동시에 가리키는 문서(co citation)와 coupling 등을 이용하여 검색결과를 범주화(clustering)하는 방법에 대해 실험하였다. 범주화효율을 높이기 위해 8개 이상의 out link를 가지고 있고, 그중 80% 이상이 같은 문서를 가리키는 문서를 복제문서로 간주하고 제거하였다. 실험결과 60 70%의 웹 문서들이 올바르게 범주화되었다. 이 기법은 링크의 정보 중의 하나인 co citations, coupling, hub, authority 등의 정보를 이용하여 클러스터링에 응용할 수 있다.

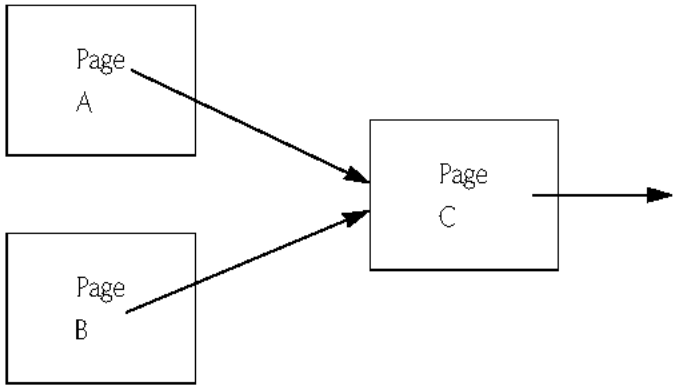
Jin과 Dumais(2001)는 링크의 정보와 content 정보를 혼합하는 알고리즘을 제안하였다. 간략하게 설명하자면, 질의어와 문서의 유사도, 링크로 연결된 문서와 문서 간의 유사도, 링크 정보를 가진 문서의 중요성(pagerank, hub, authority 점수) 등을 이용하여 내용기반 점수와 링크 기반 점수를 혼합하여 검색효율을 높이려는 연구를 수행하였다. 실험결과에서는 색인어와 PageRank를 결합한 모델이 가장 성능이 좋았으며, 색인어와 Anchor Text를 결합한 모델이 그 다음으로 좋았다. Davison(2000)은 Anchor text가 검색성능의 향상에 도움을 줄 수 있다는 사실을 보여주었다. 즉, title text, anchor text 등의 가치를 평가하는 연구를 하였다. 이 실험의 결과에서는 anchor text의 전후 단어추출은 성능향상에 크게 기여하지 못한다는 부분이 있으며, 결론적으로 anchor text가 다른 정보에 비해 유용하며, 성능향상에도 기여한다는 것을 보여주었다.

국내에서는 이상호, 강승식(2003)은 "하이퍼 텍스트의 가중치조절과 링크 구조분석기법을 통한 검색 엔진 성능개선"이란 연구에서 In link와 Out link의 Anchor text를 이용한 검색방법을 제시하고 있다.

3. PageRank 알고리즘 및 HITS 알고리즘 분석

3.1 PageRank 알고리즘

PageRank 알고리즘은 웹 검색을 위해 하이퍼텍스트의 링크 구조를 분석하는 대표적인 알고리즘이며, Brin과 Page에 의해 현재 가장 성능이 우수하다고 알려져 있는 웹 검색 엔진인 Google에서 구현되었다(Brin and Page 1998; Google 2004). 웹 문서는 다른 웹 문서로 연결하기 위한 forward 링크와 다른 웹 문서들로부터 연결되는 backward 링크들을 갖는다. <그림 1>은 보다 많은 수의 웹 문서들로부터 참조되고 있는 즉, backward 링크들을 가진 웹 문서가 적은 수의 backward 링크를 가지고 있는 웹 문서보다 중요한 문서라는 가정은 웹 문서들마다 backward 링크의 빈도수만을 고려하게 된다. 그러나 backward 링크의 웹 문서가 다른 backward 링크의 웹 문서보다 큰 중요도를 가질 경우, 즉 예를 들어, 카테고

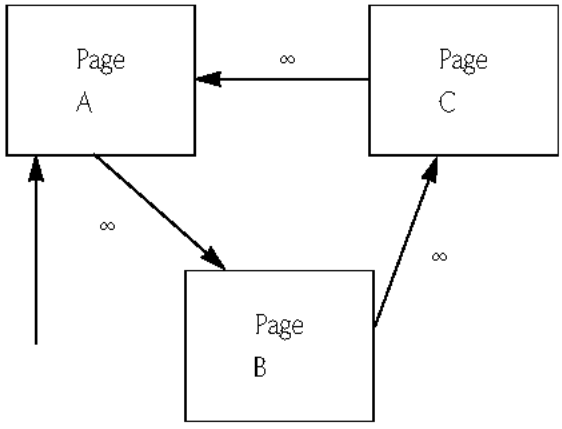


〈그림 1〉 Page C의 forward와 backward 링크

리 기반의 검색 엔진인 Yahoo로부터의 backward 링크는 다른 웹 문서들로부터의 backward 링크보다 높은 가중치를 가져야만 한다. 이로부터 PageRank 알고리즘은 링크 분석의 대상이 되는 전체 웹 문서들에서 각각의 문서들에 대한 중요도를 구하는 과정을 연속적으로 수행하여, 각각의 웹 문서들에 대한 중요도가 수렴된 값을 최종적인 중요도의 결과값으로 사용한다.

PageRank 알고리즘은 웹 문서에 대한 PageRank 값을 구하기 위해 우선 backward 링크들을 가지고 있는 웹 문서들을 찾고, 이들의 PageRank 값을 forward 링크의 수로 나눈 값들의 합을 구한다. 즉, 높은 PageRank 값을 갖는 웹 문서로부터의 backward 링크를 가질 경우, 구하려는 웹 문서의 PageRank 값에 유리하게 되며, 이것은 backward 문서의 forward 링크들의 수에 의해서 상쇄된다. 다른 웹 문서들에 영향을 주는 웹 문서의 PageRank 값은 문서가 갖는 forward 링크들에게 균등하게 나누어진다.

PageRank 알고리즘을 수행하는 것은 웹 문서의 PageRank 값이 연결되어 있는 다른 웹 문서들에 영향을 주어 PageRank 값을 증가시켜 준다. 따라서 만일 어떤 웹 문서들이 서로 연결되어져 있는 circular 그래프를 형성하게 될 때 외부로부터 연결되어져 그래프로 들어오는 링크들만이 존재하고, 그래프로부터 외부의 웹 문서들로 나가는 링크가 존재하지 않을 경우, 그래프 내의 웹 문서들에 대한 PageRank 값은 수렴하지 않고 지속적으로 증가하게 되는 "rank sink"의 문제를 갖게 된다 (〈그림 2〉). 이를 해결하기 위해 random surfer 모델을 이용하여 PageRank 알고리즘을 확장한다. Random Surfer 모델은 웹 서핑 중인 사용자가 현재의 웹 문서에 연결되어 있는 링크를 따라 이동하다가, 직접 URL을 입력하여 관련되지 않은 다른 웹 문서로 이동하게 되는 과정을 modeling한 것이다. 확장된 PageRank 알고리즘은 randomly



〈그림 2〉 Random sink problem

surfing 확률을 나타내는 상수에 의해 bacward 링크들에 대한 PageRank 값들의 합을 보정한다.

이상의 내용을 수식으로 나타내면 다음과 같다.

$$PR_{i+1}(D)=(1\ \ d)\frac{1}{N_{total}}+d\times\sum_{D_k\in In(D_j)}\frac{PR_i(D_k)}{NOut_k}$$

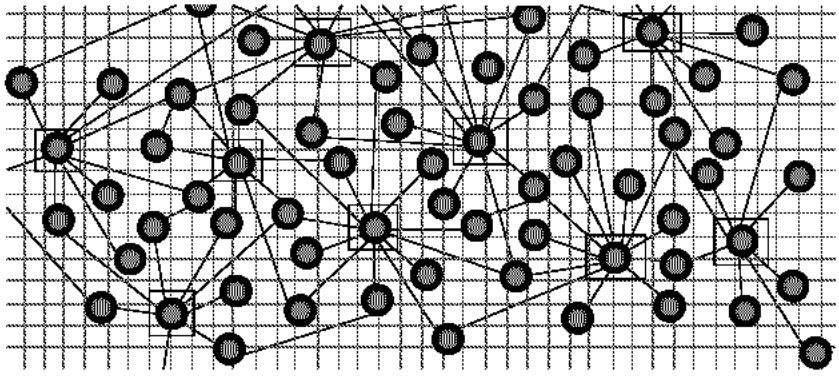
〈수식 2〉

여기서 d는 댐핑 팩터(damping factor)이며, 이는 PageRank가 수렴하는 것을 보장하기 위한 인자이다. $In(D_j)$는 문서 D_j로 향하는 링크를 가진 모든 문서들의 집합을 말하고, $NOut_k$는 문서 D_k의 전체 링크수를 나타낸다. 또한 N_{total}은 총 문서의 수를 뜻한다.

3.2 HITS 알고리즘

인터넷은 하이퍼링크를 이용하여 서로 연결된 거대한 문서의 창고이다. 문서들끼리 연결되어 있는 링크 정보를 이용한 연구는 Kleinberg의 HITS(Hypertext_Induced Topic Search)알고리즘을 계기로 많은 연구가 이루어지고 있다. Kleinberg(1998)가 개발한 HITS 알고리즘은 in link와 out link를 활용하여 Authority 페이지와 Hub 페이지를 정의하여 웹 페이지의 가중치에 적용함으로 웹 페이지에 대표성(representativeness)을 나타내었다. 이러한 페이지의 구분 중, Authority 페이지는 중요정보를 많이 내포한 페이지라 할 수 있다. 예컨대, 월드컵에 관련된 좋은 authority로는 FIFA 홈페이지가 될 수 있다. Hub 페이지는 중요정보에 대한 링크를 많이 소유한 페이지라 할 수 있다. 〈그림 3〉은 authority와 hub를 설명하고 있는데 그림에서 네모상자는 허브를 나타내며, 나머지는 authority를 가리킨다.

HITS 알고리즘은 각각의 문서마다 hub 값과 authority 값을 계산한다. 좋은 authority 값을 가지고 있는 문서는 관련 있는 내용을 가



〈그림 3〉 Authority와 Hub

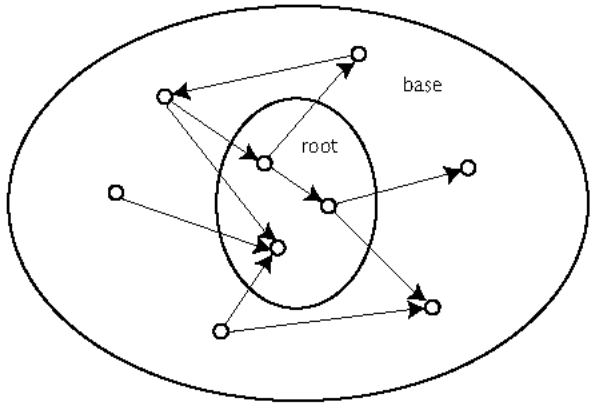
지고 있고, 좋은 hub 값을 가지고, 이 문서는 관련 있는 문서를 연결하는 링크를 가지고 있다. 좋은 authority 값을 가지고 있는 문서들에 대한 연결 링크를 많이 갖는 문서는 좀더 좋은 hub 값을 가지며, 좋은 hub 값을 가지고 있는 많은 문서들에 의해 연결되어지는 문서는 좋은 authority 값을 가지게 된다.

HITS 알고리즘은 사용자의 질의를 이용하여 내용검색을 수행한 검색결과에서 출발한다.

내용기반 검색결과를 시작집합으로 정의하고, 시작집합에 들어 있는 모든 문서들을 대상으로 문서의 링크 정보를 이용하여 1단계 확장을 통하여 HITS의 기본집합을 구한다.

HITS 알고리즘은 이러한 기본집합을 대상으로 아래와 같은 방법대로 hub 값과 authority 값을 각각 구한다.

이상과 같이 Kleinberg는 링크의 연결성을 분석하여 좋은 문서들에서 많이 연결되어 있



〈그림 4〉 시작집합을 기본집합으로 확장하는 예

(1) Let N be the set of nodes in the neighborhood graph.

(2) For every node n in N, Let H[n]be its hub score and A[n]it authority score

(3) Initialize H[n] and A[n] to 1 for all n in N

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

(5) For all n in N, A[n]:- $\sum_{(n',\ n)\in N} H[n']$

(6) For all n in N, H[n]:- $\sum_{(n,\ n')\in N} A[n']$

(7) Normalize the H and A vectors

〈수식 3〉 Kleinberg의 HITS 알고리즘

고, 좋은 문서들을 많이 연결하고 있으면 좋은 문서로 판단하였다. 그러나 문서들상에 있는 링크의 연결정보는 인터넷의 발달로 의미가 없고 불필요한 것들이 많이 발생하였다.

Bharat(1998)의 알고리즘은 Kleinberg의 HITS에 존재하는 몇 가지 문제점을 해결하고자 HITS 알고리즘에 기반을 두고 만들어졌다. Bharat은 Kleinberg의 HITS 알고리즘의 문제점을 다음과 같이 세 가지로 요약하였다.

첫째, 호스트들이 서로서로 협조적일 때 협조의 정도에 따라 특정 호스트로부터 연결되어 만 있어도 특정 호스트의 수많은 연결로 인하여 높은 hub 값과 authority 값을 가질 수 있게 된다. 둘째, 소프트웨어 산업이 발달하여 웹 문서들을 쉽게 만들 수 있는 도구를 제공하면서 문서제작도구는 특별한 의미 없이 기본적으로 문서마다 링크 정보를 추가함으로써 HITS의 계산을 의미 없게 만든다. 셋째, HITS의 시작집합이나 기본집합에 사용자의 질의에 의미 없는 문서가 좋은 연결성을 가지고 있다면, 가장 높은 hub 값과 authority 값이 사용자 질의와 상관없는 문서가 되는 topic drift 문제가

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

(5) For all n in N, A[n]:- $\sum_{(n',\ n)\in N} H[n'] \times auth_{wt(n',n)}$

(6) For all n in N, H[n]:- $\sum_{(n,\ n')\in N} A[n'] \times auth_{wt(n,n')}$

(7) Normalized the H and A vectors

(8) auth_wt(n′, n)-1/k×relevancy_wt(n′, q)

(9) hub_wt(n, n′)-1/j×relevancy_wt(n′, q)

〈수식 4〉 Bharat의 개선된 HITS 알고리즘

발생하게 된다.

Bharat은 첫 번째 문제를 해결하고자 아래와 같이 HITS 알고리즘을 수정하였다. 여기에서 사용되는 auth_wt( )와 hub_wt( )는 하나의 문서에 같은 호스트로부터의 연결이 n개 있다면 1/n의 값이 되어 하나의 호스트는 하나의 의견으로 비중을 낮췄다.

Bharat은 HITS 알고리즘의 둘째, 셋째 문제를 해결하고자 시작집합에서 상위에 랭크된 문서들로부터 확장질의를 만들었다. 이렇게 만들어진 확장질의와 시작집합에서 연결정보를 이용하여 확장된 기본집합에 속하는 모든 문서들과 유사도를 계산하였다. 기본집합에 속하는 문서들의 유사도 값들 중에 중간 값을 구하고, 중간 값 이하인 문서들은 기본집합에서 제거하고 나서 위와 같이 hub 값과 authority 값을 구하였다.

수정된 HITS 알고리즘에서는 웹 문서 n과 n′를 연결하고 있는 링크에 가중치를 부여한 뒤 authority와 hub score를 구한다. 두 호스트의 웹 문서들 사이의 링크 수로부터 보정하기 위한 인수 k는 문서 n′가 존재하는 호스트의 다른 문서들로부터 문서 n을 연결하는 링크들의 개수이고, 인수 j는 문서 n′가 존재하는 호스트의 다른 문서들로 문서 n이 연결하는 링크들의 개수이다. 두 번째 가중치 요소로서 relevancy weight는 사용자의 질의와 웹 문서 사이의 cosine distance 식으로부터 유사도를 구하여 사용한다.

이상의 내용을 정리하면 Kleingber 연구를 통해 텍스트적인 정보를 전혀 감안하지 않은, 순수하게 링크 구조만을 사용해도 상당한 수준으로 '좋은' 페이지를 판단해낼 수 있다는 사실을 알 수 있다. 특히 페이지 내부적인 정보만으로는 그 페이지가 검색결과 내에서 얼마나 '좋은' 페이지인 지를 판단하기가 힘든데, 이때 하이퍼링크를 이용함으로써 이 문제를 보다 객관적으로 해결할 수 있다는 것을 알 수 있다.

예를 들면 "Gates"라는 검색어에 대해서 상위 3개의 authority는, http://www.road-ahead.com/, http://www.microsoft.com/, http://www.microsoft.com/corpinfo/billg.html로 나타났으며, 이는 순수하게 알타비스타에서 검색한 결과에서 http://www.road-ahead.com 페이지가 123번째 결과로 나왔다고 한다(Kleinberg 1998). 또한 "search engines"라는 검색어에 대해서도 상위 5위까지를 살펴보면, http://www.yahoo.com/, http://www.excite.com/, http://www.mckinley.com/(Magellan 검색 엔진), http://www.lycos.com/, http://www.altavista.digital.com/ 이다. 이는 비교적 이용자가 원하는 정보를 잘 검색한다고 볼 수 있다.

4. 하이퍼링크 정보를 이용한 검색 엔진 사례

앞에서 설명한 하이퍼 링크 정보 알고리즘을 응용한 검색 엔진으로는 Web에서 사용 중인 Google, Ask.com을 들 수 있다.

4.1 Google

Google 시스템은 스탠퍼드 대학에서 개발 · 연구하고 있는 검색 엔진으로서 웹 페이지의 하이퍼링크와 하이퍼텍스트를 분석하여 이를 기반으로 페이지의 가중치를 계산하였다. 이때 전체 웹에서 모든 페이지를 수집하여야 하는데, 방대한 양의 페이지를 수집할 때 저장 공간과 시간을 고려하여 선택적으로 페이지를 선별, 저장하는 방법을 제안하고 실험을 통한 여러 가지 방법으로 문서를 수집하였다. 질의어와 페이지들 사이의 벡터 유사도(vector similarity)를 사용하는 방법, 백 링크(back link), 포워드 링크(forward link), PageRank(pagerank) 방법 등 다양한 방법을 적용하였다. 이렇게 수집된 페이지들의 가중치를 계산하는 방법으로는 하이퍼링크를 이용하는 PageRank 방법과 하이퍼텍스트로 가중치를 계산하는 방법을 같이 사용하였다.

페이지 랭크 방법은 다음과 같은 식을 사용하여 페이지의 가중치를 계산하였다(Brin and Page 1998).

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

또는

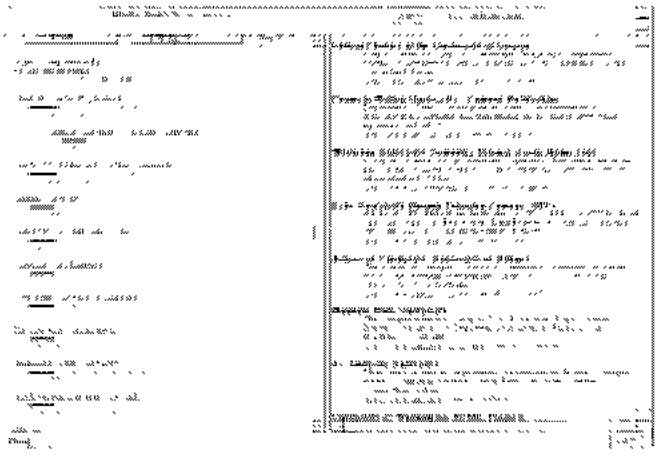
$$PR(A) = (1\ \ d)/N + d(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

여기서 d는 감소상수로서 보통 0.85를 사용하여 계산하였다. PR(A)는 페이지 A의 PageRank 값이고, T1, ... Tn은 A로의 링크가 있는 페이지이며, C(A)는 페이지 A 안에 있는 하이퍼링크의 수이다.

하이퍼텍스트는 하이퍼링크와 같이 연결되어져 있는 텍스트이다. 가령 〈A HREF="..."〉 text 〈/A〉의 링크가 있을 경우 text를 의미한다. Google 검색 엔진에서는 위에서 살펴 본 PageRank 값과 하이퍼 텍스트에서 질의어가 존재하는 빈도 등을 이용하여 페이지의 가중치를 계산하였다.

링크 구조와 링크 텍스트는 "적합성 판단(relevance judgement)"과 질적인 필터링에 있어서 많은 정보를 제공할 수 있다. 구글은 그 링크 구조와 앵커 텍스트를 사용하고 있다.

구글 검색 엔진은 개별 웹 페이지의 품질을 순위 매기기 위해 웹의 링크 구조를 이용한다. 이 랭킹은 페이지 랭크라 일컬어진다. 둘째, 구글은 검색결과를 개선하기 위해 링크를 사용한다. 구글은 PageRank 외에도 앵커 텍스트를 이용한다. 일반적으로 검색 엔진은 링크의 텍스트 그 자체를 특별하게 다루며, 링크의 텍스트를 링크를 담고 있는 그 페이지와만 연관시킨다. 그러나 구글은 여기에 더 나아가 링크가 가리키는 페이지까지 링크의 텍스트와 연관시킨다. 이것은 여러 가지 장점이 있다. 첫째, 앵커는 자주 그 링크가 담겨있는 페이지보다 그 링크가 가리키는 페이지에 대한 보다 정확한 설명을 담고 있는 경우가 많다. 둘째, 일반적인 텍스트 검색 엔진이 인덱싱할 수 없는 이미지나 프로그램, 데이터베이스로의 앵커(링크)도



〈그림 5〉 구글 및 일반검색 엔진의 검색결과 화면

존재할 수 있다. 그러므로 앵커를 활용하면 실제로 크롤링되지 않은 웹 페이지들까지 찾아낼 수 있다.

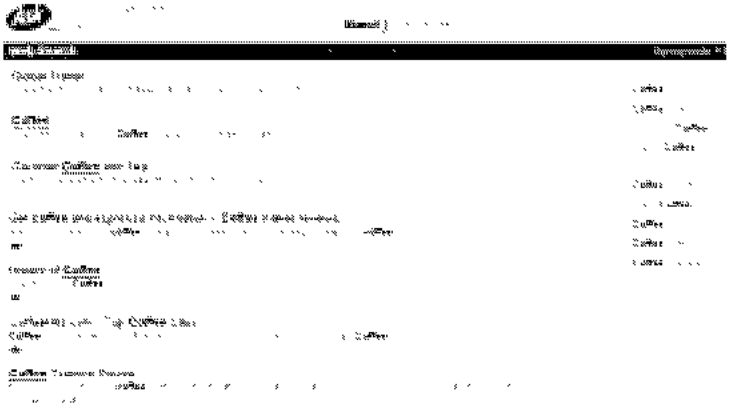
PageRank 기술과 앵커 텍스트를 사용한다는 점 외에 구글은 다음과 같은 특징을 가진다.

첫째, 구글은 모든 히트(hit)에 관한 위치정보를 저장하기 때문에 검색 시 근접도를 광범위하게 활용한다. 둘째, 구글은 어떤 단어의 폰트 크기와 같은 시각적인 세부요소를 추적한다. 폰트 사이즈가 큰 단어나 볼드체로 된 단어는 그렇지 않은 단어에 비해 더 높은 가중치가 부여된다. 셋째, 구글은 완전한 HTML도 저장하기 때문에 이를 이용할 수 있다.

이상과 같이 구글은 링크 구조를 이용한 검색순위 방법을 이용하고 있으며, 이들은 실제로 "university"라는 질의어를 구글과 일반검색 엔진인 알타비스타에 주어서 검색한 결과 〈그림 5〉와 같이 나타났다. 그림에서 보듯이 일반검색 엔진은 university라는 단어가 많이 들어간 페이지들을 보여준 반면 구글은 실제 대학 홈페이지들이 대부분 올라와 있다는 것을 알 수 있다(Brin and Page 1998).

4.2 Ask.com

Ask.com은 IAC Search & Media의 다양한 서비스 중의 하나로 1996년 David Warthen, CTO, Garrett Gruener 등에 의해 Berkely에서 창설되었다. 현재 IAC Search and Media는 웹 기반 검색 관련 브랜드로 Ask.com, Ask for Kids, Bloglines, Exite, Excite, iWon, FunWebProducts, MyWay 등 8개를 소유하고 있다. 이 IAC Search and Media 회사는 2001년에 Teoma 검색 엔진을 인수했으며, 2006년 2월에 AskJeeves를 Ask.com으로 이름을 변경하였다. 현재 Ask.com에서 사용하는 검색 알고리즘은 Teoma 검색 엔진과 AskJeeves의 검색 알고



〈그림 6〉 Ask.com의 검색 화면

리즘을 종합한 기술인 ExpertRank라는 알고리즘이다. Ask.com의 기본적인 아이디어는 자연언어로 질의를 받아서 그에 해당하는 결과를 보여주는 것이었다. Ask.com은 일반적인 키워드 검색뿐만 아니라 자연언어(영어)로 이루어진 질의에 답변할 수 있고, 다른 검색 엔진들에 비해서 더 사용자에 친숙하고 직관적으로 접근하고 있다. 또한 웹 사이트의 권위를 판단하기 위해서 특정 주제에 관련된 권위 있는 페이지들의 링크를 사용하는 Teoma라는 검색 엔진 기술을 보유하고 있다. 현재 Ask.com에서 사용하고 있는 Teoma 검색 엔진은 클러스터링을 통해 query expansion(해당 질의어와 관련된 확장된 질의어 추천)을 가능하게 해준다. 〈그림 6〉은 Ask.com의 질의어 상자에 'coffee'라고 입력한 후 검색된 결과이다. 이 그림의 오른쪽 부분에 보면 coffee로 검색했을 때 해당 질의어와 가장 많이 관련 있는 한 단계 더 깊은 질의어들을 추천해주고 있다.

사실 Teoma는 2000년에 Apostolos Gerasoulis 교수와 그의 동료들에 의해서 Rutgers University에서 만들어진 검색 엔진인데, 이 검색 엔진에 대한 연구는 1998년의 DiscoWeb 프로젝트로부터 시작하였다. 원래의 연구는 "DiscoWeb: Applying Analysis to Web Search."라는 논문에 발표되었었다. Teoma는 link popularity 시스템이라는 특이한 기능을 갖고 있다. Google의 PageRank와는 다르게 ExperRank의 기술(Subject Specific Popularity)은 어떤 웹 사이트 간의 연결관계를 특정 주제와 관련해서 분석을 한다. 예를 들어, 'baseball'에 대한 웹 페이지는 다른 'baseball'에 대한 웹 페이지들이 그 페이지로 연결이 되었을 때 순위가 높아진다(Google의 PageRank는 주제와 상관 없이 Page 자체의 Authority를 본다.).

이상과 같이 현재 웹 검색을 위해 하이퍼텍스트의 링크 구조를 이용하고 있는 대표적인

알고리즘인 PageRank와 HITS에 대해 살펴보았는데, 이들의 공통점은 모두 검색순위결정에 하이퍼링크를 이용한다는 점이다. 그리고 Kleinberg의 HITS 알고리즘이나 PageRank 알고리즘은 링크 분석의 대상이 되는 전체 웹 문서들에서 각각의 문서들에 대한 중요도를 구하는 과정을 연속적으로 수행하여, 각각의 웹 문서들에 대한 중요도가 수렴된 값을 최종적인 중요도의 결과 값으로 사용하고 있다. PageRank 알고리즘의 특징은 검색과 독립적인 랭킹 시스템을 사용한다는 점이다. 반면에 HITS 알고리즘은 검색 시 적합도계산에 필요한 연산량이 많아 작은 문서집합에 대해서만 적용할 수 있었다. 그러나 PageRank는 웹으로부터 가져온 문서들에 대해 미리 문서의 적합도를 계산해 놓고, 검색 시 이 값을 참조하여 정리해 결과를 보여주므로 검색속도가 빠르고, 거대한 웹 문서량을 검색할 수 있다는 장점이 있다. 이 점이 구글에서 실제 검색 엔진에서 성공적으로 응용할 수 있었던 요인이라 할 수 있다. 그러나 문제는 날로 늘어나는 문서수로 인해 검색결과, 중요도의 품질이 점점 손상되고 있다는 점이다. 더욱이 의도적으로 중요도를 높이려는 시도가 늘어나며, 사이트 간 배너 교환이나 링크만 가지고 있는 다량의 스팸 문서를 생성해 인위적으로 랭크를 높이는 경우가 많아 구글의 페이지 랭크 시스템에 혼란을 주고 있다. 이밖에도 검색최적화기법(Search Engine Optimization)이 발달하면서 순수하게 문서의 인기도와 중요성에 따라 링크가 생성되던 기존 웹 문서의 하이퍼링크 네트워크 구조가 점점 왜곡되어가고 있는 것이 사실이다. 이에 따라 Ask.com과 같은 검색 엔진들은 구글의 랭킹 시스템과 차별화된 새로운 랭킹 시스템을 내세우며 업계진출을 꾀하고 있다. Ask.com은 같은 주제 간 링크에 가중치를 주는 방식으로 적합도의 품질을 높여 구글에 도전하고 있다.

5. 결 론

정보검색은 90년대 이전에는 단어 출현빈도 등과 같은 해당 문서의 내용을 기반으로 문서를 검색하는 방법이 주류를 이루었으며, 90년대에는 링크 정보를 이용한 연구가 많이 진행되었다. 이런 링크를 이용한 연구는 90년대 후반에 접어들어 HITS와 PageRank가 링크를 이용한 대표적인 사례로 인정되었다. 따라서 본 연구에서는 HITS와 PageRank 알고리즘에 대해 살펴보고, 이런 알고리즘 기법을 적용하고 있는 검색 엔진인 Google과 Ask.com에 대해 살펴보았다.

인터넷상에 분포되어 있는 웹 페이지에 대한 효율적인 검색 서비스를 수행하기 위해서는 웹 페이지들의 검색결과를 순서대로 화면에 디스플레이시키기 위해 효과적인 순위매김(ranking)이 필요하다는 것은 이미 잘 알려져 있다.

현재 서비스되고 있는 대부분의 검색 엔진은 기본적으로 사용자가 입력한 검색어와 일치

하는 단어를 포함한 문서를 검색한다. 그리고 이런 매칭 검색에 의한 검색결과를 내부적인 기준에 의해 결정된 정확도, 또는 중요도에 따라서 다시 정렬을 한 결과를 이용자에게 보여준다. 그런데 웹 문서 검색 엔진의 경우는 그 특성상 인덱싱하고 있는 문서가 매우 많기 때문에 보통 검색결과가 수만에서 수십만에 달하며, 검색 엔진이라면 수백만개의 검색결과를 출력하는 경우도 있다. 그러나 대부분의 경우에 있어서 사용자가 원하는 정보는 검색결과의 극히 일부에 지나지 않는다. 따라서 사용자에게 필요한 정보, 원하는 정보를 검색결과의 상위에 출력할 수 있는 기능이 웹 문서 검색 엔진에서는 매우 중요하다.

예를 들어, 자동차 이름과 고양이과 동물의 이름이란 중의적 의미를 갖는 Jaguar로 검색을 할 경우, 자동차 재규어에 대한 웹 문서가 고양이과 동물인 재규어에 대한 문서가 차지하는 현상이 나타난다. 또한 상위에 랭크된 페이지들이 각자 다른 정보를 제공하고 있는 것이 아니라, 대부분 비슷한 문서들이기 때문에 좋은 검색결과라고 생각하기 힘들다. 따라서 유용하면서 동시에 다양한 정보를 상위 랭크에 제공할 수 있고, 동시에 검색 시 추가연산이 필요하거나 지나치게 많은 저장공간을 필요로 하는 문제를 극복한 실용가능성 있는 랭킹 알고리즘을 고안할 필요가 있다. 이러한 정확한 랭킹 시스템을 구현하기 위해서는 검색어의 빈도수, 주제의 특성, 링크의 빈도수, 링크 텍스트의 반영정도, 스팸 문서의 필터링 정도를 통해서 검색대상의 신뢰도를 높일 수 있는 방안을 모색해야 할 것이다.

참고문헌

이상호, 강승식. 2003 하이퍼텍스트의 가중치 조절과 링크 구조 분석기법을 통한 검색 엔진 성능개선. 『제 15회 한글 및 한국어 정보처리 학술대회』. 2003년 10월.

Agichtein, E. S, Lawrence, and L. Gravano 2001 "Learning Search Engine Specific Query Transformations for Question Answering." *In Tenth International World Wide Web Conference*, May, at HongKong.

Sergey Brin and Lawrence Page 1998. "The Anatomy of a Large scale Hypertextual Web Search Engine." *Proceedings of the Seventh International Conference on World Wide Web* 7, 107 117.

Bharat, K. and M. Henzinger 1998. "Improved Algorithms for Topic Distillation in a Hyperlinked Environment." *ACM SIRIR*.

Chakrabarti, S., et al. 1999. "Mining the Web's Link Structure," *IEEE computer*, 32(8): 60 67.

Chakrabarti, S., B. Dom, P. Raghavan, S. Rajagopalan, et al. 1998. "Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text," *Computer Networks and ISDN Systems Archive*, 30(1 7): 65 74.

Craswell, N., D. Hawking, and S. Robertson. 2001. "Effective Site Finding using Link Anchor Information" *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 250 257.

Davison, D. 2000. "Topical Locality in the Web," *In Procs of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 272 279.

Filman, R, E. and S. Pant. 1998. "Searching the internet" *IEEE Internet Computing*, July, Aug: 21 23.

Google. 2004. 〈http://www.whitehatseo.org/guidelines/google_en.pdf〉. [cited 2005. 5].

Dumais, C. 2001. "Probabilistic Combination of Content and Links," *Proc. of ACM SIGIR*, 01: 402 403.

Kleinberg, M. 1998. "Authoritative Sources in a Hyperlinked Environment," *The Journal of the ACM*, 46(5): 604 632.

Schwartz and Candy. 1998. "Web Search Engines," *Journal of the American Society for Information Science*, 49(11): 973 982.

Wang, Y. and M. Kitsuregawa. 2001. "Link Based Clustering of Web Search Results," *Second International Conference on Advances in Web Age Information Management* (WAIM): 225 236.

〈http://www.google.com〉. [cited 2006. 5].

〈http://www.ask.com〉. [cited 2006. 5].