정보학연구 제1권 제2호 1998년 11월(pp.55-64)

검색엔진 성능의 정량적 분석

조석팔
성결대학교 전산정보학과 조교수

요 약

본 논문은 웹 상에서 하이퍼텍스트 문서의 정보 검색에 있어서 검색에 요구되는 질의어에 따른 검색 결과가 주제에 따른 관련성을 측정하며, 하이퍼텍스트 문서가 링크되는 문서 상호간의 유사성에 대하여 정량화를 시도함으로써 검색 엔진의 성능분석을 제시한다

Ⅰ. 개 요

하이퍼텍스트는 기본적으로 정보를 접근하는데 있어서 비순차적이고 탁월한 방법을 제공하는 데이터베이스 시스템이다. 그리고 필수적인 기능은 하이퍼링크와 노-드이다. 노-드는 텍스트, 그래픽, 오디오, 비디오 및 다른 미디어를 포함한다. 하이퍼링크 노-드와 접속되어 정보의 비 선형적인 조직 구성을 지원한다. 오늘날 가장 인기 있고 확장된 하이퍼텍스트 시스템은 웹이다. 그러나 전통적인 문서 전용 데이터 베이스와는 달리 여기서의 각 문서는 전체와 관련하여 기술되어 지거나 선택되어지며, 개방 환경의 인터넷의 웹 상에서 문서를 발송할 수 있다. 자원이 인터넷상에서 존재한다는 사실은 그 자원의 중요성, 정확성, 이용성 및 가치 등을 보장한다고 할 수 없다[1].

논문접수일 : 98년 7월 게재확정일 : 98년 10월

대부분의 인터넷 검색은 관심 있는 주제에 대하여 일반 사용자가 단순한 질의를 함으로써 시작된다고 할 수 있다[2]. 그러한 질의는 간단한 클릭에 의하여 수천 개가 발생할 수 있기 때문에 질의에 대한 문서의 관련성을 나타내는 등급이 검색 엔진에 있어서 핵심 기술이다. 전통적인 정보검색 이론은 사용자 정의에 의한 키워드와 문서 내용과의 상호 유사성을 측정하는 모형을 제공하는 것이었다. 이러한 모형은 벡터 공간 모형, 확률적 모형, 및 퍼지 논리모형 등을 포함한다[3]. 대부분의 모형은 주어진 문서에 대한 질의 항목의 빈도에 의존한다. 그러나 이러한 접근은 인터넷상에서의 경우가 아닌 데이터베이스의 문서상에서의 완전하고 무결함을 전제로 한다. 이와는 상반하여, "키워드 검색"은 검색결과의 등급을 올리기 위해 웹 현장 경영자가 사용하며, 여러 가지 경우에 있어서 단어가 문서에 나타나는 빈도에 따라서 내용의 질을 판단하지는 못한다.

이 항목은 하이퍼링크 벡터 결정 방법, 하이

퍼텍스트 문서 검색 및 순서 매김에 대하여 새로운 방법을 기술한다. 하이퍼텍스트 벡터 결정 방법은 질의 항목에 대한 관련성에 등급을 매기기 위해서 하이퍼링크의 내용을 사용한다. 따라서 하이퍼링크 벡터 결정은 과학적 인용 목차와 같이 동작한다. 여기에 주어진 자료를 인용하는 자료의 수는 그 자료의 내용의 질 과 주제에 관련성을 측정하는데 있어서 효과적인 도구로 검증되었다. 하이퍼링크 벡터 결정에 있어서 현장에 대한 하이퍼링크는 자료에 대한 인용이다.

하이퍼텍스트 구현에 대한 그들의 관련과 몇 가지 기본 정보 검색 개념을 소개한 후, 하이퍼링크 벡터 결정 방법을 간단한 검색엔진으로 실험한 결과에 따라 좀도 상세히 기술하고자 한다.

II. 정보검색

정보 검색 시스템에서, 문서는 이를 발송하는 각 항목을 기록하는 변환 목차를 생성하기 위하여 전처리 되어진다. 발송은 항목, 문서 식별자, 그리고 문서에서 그 항의 가중치를 부여하는 요소이다. 항목의 발송은 문서 식별자에 의하여 분류되어진다. 유사성에 기본을 둔 검색 알고리즘은 질의어와 유사한 항목을 가지는 문서의 관련성에 대하여 점수를 계산하기 위하여 각 질의 항목에 대한 발송 횟수를 참조한다.

벡터 공간모형은 문서 나 질의, 또는 두 문서 사이에 유사성을 정량화 하기 위하여 정보 검색에서 폭넓게 사용되어진다. 이 모형은 문서가 수집되는 어휘에 따라 각 유일한 항목에 서 하나의 소자성분을 가지는 벡터에 의하여 표현되어진다. 각 소자 성분의 가치는 문서상에서의 각 항목별 및 빈도 수에 따라 가중치가 부여되어진다. 문서상에서 일어나지 않는 항목은 가중치 "0"을 가진다. 질의는 벡터로 표현되어지며 질의와 문서와의 사이에 유사성은 그들 항목 벡터의 내부 곱으로서 계산되어진다. 이러한 측정은 두 벡터사이의 코사인 각과 동일하여 가끔 코사인 유사성으로 호칭되어진다.

예를 들어서 벡터 공간 모형에서 사용자 질의 $\bar{I}$ 는 벡터이며 질의에서 각 키워드는 질의 벡터의 크기는 식(2-1)과 같이 나타낼 수 있다. 여기서 n는 질의에 있어서 키워드의 번호이다. 인터넷 검색 엔진에서 모든 질의의 70%는 n값이 1 혹은 2이나 그이상의 큰 값을 가지는 경우도 있다.

$$\bar{I} = (i_1, i_2, \ldots, i_n) \qquad (2\text{-}1)$$

문서 $\bar{P}$ 는 벡터로 표현되어지며, 기본적으로 k는 문서 $\bar{P}$에 있어서의 키워드 번호로서 식(2-2)와 같이 표현할 수 있다..

$$\bar{P} = (p_1, p_2, \ldots, p_k) \qquad (2\text{-}2)$$

관련성에 관한 점수는 사용자 질의 $\bar{I}$ 와 문서 $\bar{P}$를 동일한 크기로 정규화 한후에 내적(內積)으로 식(2-3)과 같이 계산되어진다.(여기서 $\bar{P}$에 부합되는 키워드가 $\bar{I}$에 속해지지 않으면 부합되는 크기는 $\bar{I}$내에서 0의 값을 가진다.)

$$R = \bar{I} \cdot \bar{P} \qquad (2\text{-}3)$$

$\bar{I}$ 나 $\bar{P}$에 있어서 크기의 계산은 항목가중치로 계산되어지며 가중치 계산은 식(2-3)과 같다.

$$V_{w,q} = f_{w,q} \cdot \log(N/f_q) \qquad (2\text{-}3)$$

여기서 $f_{w,q}$는 질의 또는 문서인 q에서 단어 w의 발생횟수를 나타낸다. 그리고 N는 수집에 있어서 문서의 수이며 f_q는 q항목을 포함하는 문서의 수이다.

이 기능은 이러한 단어들이 공통된 단어보다도 더 많이 식별된다는 가정 하에서 희소한 단어는 높은 가중치를 할당한다. 다른말로 말해서 질의 와 문서에 있어서 희귀한 단어의 발생은 높은 관련성을 가진 지시자로 가정하며 이러한 가정은 웹처럼 하이퍼텍스트 시스템에서는 고정되어있지 않다. 그러나 하이퍼텍스트 시스템은 하이퍼텍스트 문서가 그들의 의미 내용을 나타낼 뿐만 아니라 다른 문서에 대한 하이퍼링크를 표현한다.

하이퍼텍스트의 문맥에 있어서 상관성 연구는 노-드 사이에 사용자 정의의 링크에 중점을 두어 하이퍼텍스트 망[6][7]을 브라우징하는 질의 기반의 전략으로 집적 화하는 노력일 수도 있고 접속된 노-드[8]와 하이퍼링크를 사용하여 검색 성능을 개선하기 위한 실험일 수도 있다. 망 구성에 있어서 여러 유형의 링크로 작용하는 증거를 상호 결합함으로써 검색의 효과를 개선시키는 작용을 한다[9].

그러나 이러한 하이퍼텍스트 프로젝트는 문서의 내용에 따라서 일차적으로 검색결과에 등급을 부여함으로써 전통적인 상관성을 적용하며, 추가 정보로 하이퍼링크를 고려하여 관련성 등급이 보다 낮은 가중치로 부여한다. 더욱이 하이퍼텍스트 상관성은 등급 별로 접속된 노-드에서 문서내용에 의한 요인을 연구한다. 이러한 추가적인 문서는 비록 특수한 상황에서 효과적일 수 있지만 일반적으로 비효과적인 지점에 등급을 부여하는 것은 복잡하다. 예를 들어서 추가적인 정보를 위해 접속된 노-드를 사용하여 수집의 량이 적은 의료나 특수학문과 같은 특별한 분야에 속해있을 때 검색의 질을 개선할 수 있다.

Ⅲ. 하이퍼링크 벡터 결정 방법

하이퍼링크 벡터 결정 방법은 하이퍼링크의 수에 기초하여 문서에 등급을 부여하며 하이퍼링크 종합 문서를 문서의 내용처럼 사용한다. 필요에 따라서, 데이터베이스가 충분히 클 때 검색결과는 "하이퍼링크 벡터결정"결과와 유사하다. 문서의 내용은 저자가 어떻게 작성하느냐에 따르는 것 보다는 어떻게 기술하느냐에 기초를 두어 선택되어진다. 하이퍼링크 벡터 결정 방법은 문서를 기술하기 위해 링크 벡터를 사용하며, 각 링크 벡터는 문서를 지시하는 하이퍼링크를 기술한다. 다시 말해서 하이퍼링크의 주요 관리자는 문서의 URL이다. 하이퍼링크의 꼬리-즉 연결 문서는 문서로 처리되어지며 그리고 연결문서의 각 항목의 가중치는 문서에 대한 하나의 링크벡터로 정의한다.

순차적으로 연결되는 동안, 순차 엔진은 스파이더를 사용하여 전체 문서의 데이터 베이스를 왔다갔다하며, 각 문서에 대한 하이퍼링크 정보를 수집하고, 다음과 같은 형태의 리스트를 생성한다.

문서식별자 : anchor-text

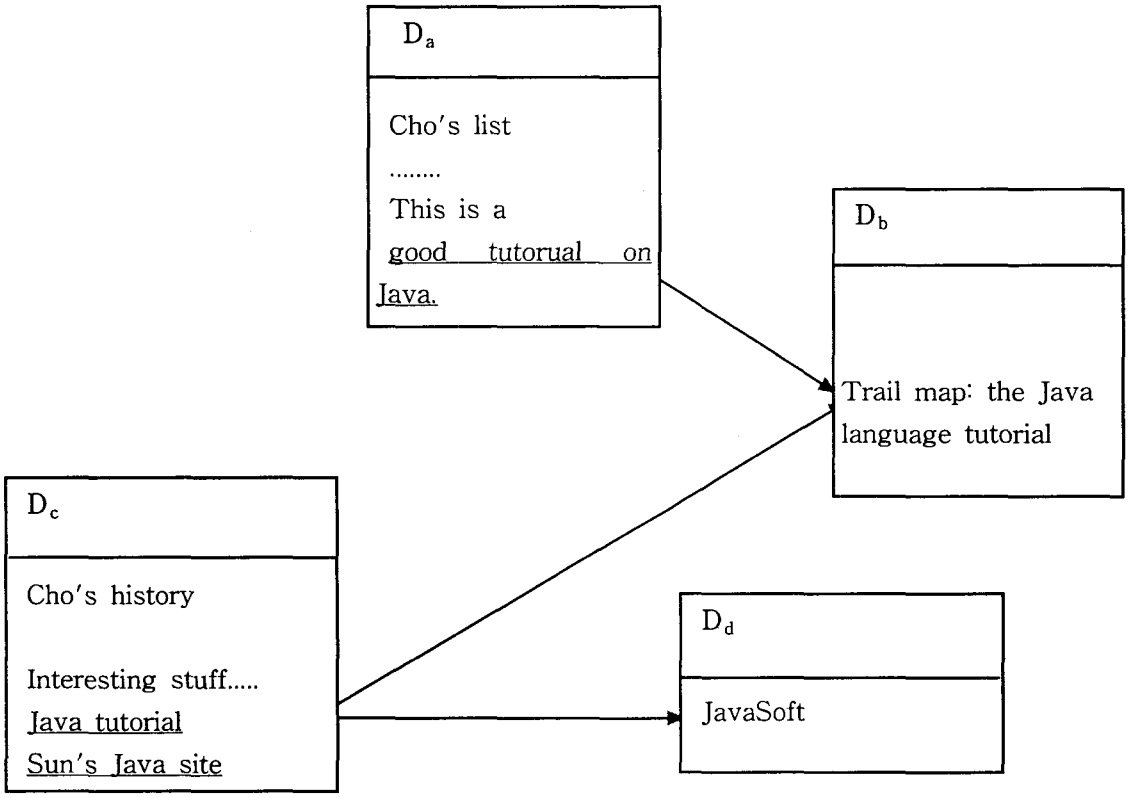
여기 문서 식별자는 하이퍼 링크의 헤드 앵커이다. 검색엔진은 이러한 정보로부터 변환된 순서를 재생 할 수 있으며, 기본적인 형태는 다음과 같다.

항목: $DF, D_1, D_2, \ldots\ldots, D_i, \ldots\ldots, D_{DF}$

여기 항목은 anchor-text로 부터의 항이며, DF는 항목에 대한 문서의 빈도이다, 그리고 D_i는 anchor-text에 연결된 i 번째 문서이다. 최종적으로 하이퍼텍스트 문서는 다음 식(3-1) 같이 링크 벡터로 표현되어진다.

$$D_j = \begin{bmatrix} \overline{L_1} \\ \overline{L_2} \\ \overline{L_3} \\ \cdots \\ \overline{L_i} \end{bmatrix} \quad (3\text{-}1)$$

여기 D_j 는 문서 식별자이며, $\overline{L_i}$ 는 헤드 앵커가 D_j인 i 번째 하이퍼링크 벡터이다. 각 링크벡터 크기의 값은 항목별 가중치를 부여하는 방법



으로 계산되어진다. 그림 1은 4개의 노드 (D_a, D_b, D_c, D_d)와 3개의 하이퍼링크 (D_a, $\rightarrow D_b$, D_c, $\rightarrow D_b$, $D_c \rightarrow D_d$)를 나타낸다.

그림 1. 하이퍼텍스트 시스템, D_a에서 하이퍼링크 헤드앵커는 D_b에 대한 URL이며 꼬리-앵커[및 하이퍼링크 내용]는 "good tutorial on Java" 이다.

이러한 과정을 왕복하는 동안 링크 정보는 다음과 같은 내용을 추출 할 수 있다.

D_b : "good tutorial on Java";
"Java Tutorial"
D_d : " Sun's Java Site"

good	1	D_b	
tutorial	1	D_b	
on.	1	D_b	
java	2	D_b,	D_d
sun's	1	D_d	
site.	1	D_d	

문서에 대한 하이퍼링크 벡터 표현은 다음과 같다.

good tutorial on java
D_b : < 2, 2, 2, 1 >
java tutorial
< 1, 2 >
sun's java site
D_d : < 2, 1, 2 >

대부분의 검색 엔진에 있어, 정보 검색 동안에 하이퍼링크 벡터 결정 방법은 관련 문서를 위치시키기 위하여 변환 인덱스와 질의어를 정합 시킨다. 그래서 하이퍼링크 벡터 결정은 하이퍼링크 벡터 표현에 근거한 문서에 등급을 부여한다. 링크 기술과 유사하게 질의는 벡터로 표현되어지며 등급 점수는 주어진 문서에 대하여 질의 벡터와 각 하이퍼링크 벡터사이에서 내적(內積)의 합으로 정의되어지며 합하는 과정은 벡터결정과 유사하다. 즉 보다 많은 링크가 기본적으로 보다 많은 점수를 받게된다. 그러나 벡터결정은 가중치가 부여되어지며 링크와 질의 벡터 사이의 유사성에 종속되고 등급의 수식은 식(3-2)와 같이 나타낼 수 있다.

$$R = \sum_{i=1}^{n} (\bar{I} \cdot \overline{L_i}) \qquad (3\text{-}2)$$

여기 R는 등급 점수이며, $\bar{I}$ 는 질의 벡터이고 $\overline{L_i}$ 는 링크벡터이다. 그림 1에 하이퍼텍스트 시스템에서 질의가 "Java Tutorial"이면 다음가 같은 벡터로서 표현되어진다.

java tutorial
질의 : <1, 2>

비슷하게, D_b에 대한 문서 링크 벡터를 다음과 같이 계산 할 수 있고 D_b와 D_d에 대한 등급 점수는 각각 1.62 와 0.149이다. 문서 a 나 c 에 대하여 하이퍼링크가 지시되지 않기 때문에 비록 두 개의 문서가 "Java"와 "tutorial" 이라는 단어를 포함하고 있다고 하여도, 질의 "Java tutorial에 대한 등급점수는 "0"이다. 하이

퍼링크 벡터 결정은 두 문서에 대하여 동일한 비율을 할당 할 때, 기존의 순차 매김과 검색 방법은 좀더 논의되어질 수 있다. 이 경우 문서 a와 c에 대한 기존의 관련성 점수는 D_b, D_d, D_c, D_a의 최종의 관련성 등급을 제공하면서 0.8과 0.6으로 각각 되어진다.

Ⅳ. 실험 결과

http://rankdex.gari.com에서 가능한 하이퍼링크 벡터 결정을 기본으로 한 실험적 웹 검색 엔진 "랭크덱스"[10]를 사용하여 시도하였다. 하이퍼링크 벡터 결정의 순차 매김 구조에 따라 530만개의 하이퍼텍스트 문서를 인터넷상에서 수집하고 순차 매김을 하였다. 순차 매김의 전체의 규모는 100메가바이트 정도이다.(기본적으로 500메가바이트 정도 되는 기존의 인덱스보다도 적다.)

웹 문서에 대한 질의 세트에 대한 진가(眞價)를 가지고 있지 않기 때문에 실험적 시도를 하였다. 일반적으로 사람에 의하여 검토되고 관련성으로 판단된 문서는 사용자 질의의 진가에 근접하는 것으로 받아들여진다. 그러므로 인기있는 질의 세트를 모아서 수동적으로 "편집자 검색 엔진이라고 불리워지는 웹 문서를 부분별로 범주를 구분하고 검토하는 검색 엔진으로 송부하였다. 동일한 질의를 "랭크덱스"에 보내고 "랭크덱스"의 상위 10개 질의에 대하여 얼마나 많은 것이 편집자 엔진에 의하여 돌아오고 관련성으로 선택되어 졌는가를 계산했다. 비교에 있어서 다은 주요 검색 엔진으로 질의를 보냈고 그들의 상위 10 개질의에 대하여 얼마나 많은 것이 편집자의 검색 엔진에 의하여 선택되어 졌는가를 계산했다.

사용자 질의에서 규정된 키워드의 평균수는 1.5이기 때문에 10개의 짧은 검색질의의 표현 단위를 사용하였다[11]. 많은 검색 엔진이 편집자 선택의 집합과 편집자에 대한 자신들의 팀을 가지고 있었다. 많은 주요 검색엔진과 "랭크덱스" 와 비교하였기 때문에 편집자 엔진에 있어서 적게 알려진 "룩크-스마트"(http://www.looksmart.com)과 같은 것을 선택하였다.

<표 1>은 검색의 결과를 보여주고 있다.

각각에 대하여 상위10개 질의 결과를 시험하면서 6개의 검색에 대해 10개의 질의를 사용하였다 "룩크-스마트" 편집자는 10개의 질의에 대하여 79 웹사이트를 선택한다. "랭크덱스"는 "룩크-스마트"의 검색 결과의 22.7% 정도인 18개가 선택되며 가장 근접한 상업적 검색엔진인 "익사이트"는 6.3%인 5개가 선택되어진다. 그리고 나머지 "인포시크"는 5%인 4,"알타비스타"는 1.2%인 1이고, "라이코스"는 0이다. 하이퍼링크 벡터 결정은 대부분 인위적인 노력으로 시뮬레이션을 수행하였으며 평가된 다른 검색 엔진보다는 사용자의 정보 요구를 만족한다.

"룩크-스마트"에 의하여 돌아온 79문서들은 완전한 관련성을 나타내고 있지 않았다. 예를 들어서 질의 "virtuality"는 상위 10에 속하였다. 분명한 것은 낮은 선택 퍼센트라고 하여 검색엔진이 "룩크-스마트"보다 나쁘다는 것이 아니라 선택 문서가 질의 에 대하여 얼마나 관련성이 있나 에 따라 품질이 좌우된다.

"랭크덱스"의 순차 매김은 다른 검색엔진 순차 매김 보다도 뚜렷이 적다. 즉 2000만~5000만과 비교 할 때 530만 페이지 정도이다. 하이퍼링크 벡터 결정방법을 큰 하이퍼텍스트 시스템

에 적용하는 것을 검증하기 위하여 다른 시험을 시도하여 보았다. 검색 엔진 시험에서, 우선 다

〈표 1〉 검색 결과 비교

질 의	룩크-스마트	랭크덱스	알타비스터	익사이트	인포시크	라이코스
communication	10	2	0	1	0	0
skating	10	2	0	1	0	0
virtuality	10	1	0	1	0	0
audio	5	3	0	0	2	0
video	8	2	1	0	1	0
multimedia	10	1	0	0	0	0
seoul	9	1	0	0	0	0
airlines	4	2	0	1	0	0
animal	10	2	1	0	0	0
market	3	2	0	1	1	0
Total	79	18	2	5	4	0

른 검색 엔진으로부터 N번 검색 결과를 가졌으며 순차 매김과 검색에 하이퍼링크 벡터 결정을 적용하였다. 하이퍼링크 벡터결정 방법을 사용하여 관련성 점수를 계산하고 하이퍼링크와 앵커 텍스트를 추출하였다. N 문서들에 재 등급을 부여하였으며 새로운 상위 10 질의 리스트와 "룩크-스마트"검색 결과와 중첩하여 비교하였다.

기본 검색엔진에 따라서 N=500으로 하여"익사이트"를 사용하였다. 표 2에서 앞에서 사용하였다. 동일한 질의 세트를 사용한 결과를 보여주고 있다.

"익사이트-벡터"는 첫 번째 500질의로 적용되어질 때 "룩크-스마트"와 중첩하여 보여주고 있다. 비교에 있어서 "랭크덱스"와 "익사이트" 만 리스트 되어진다. "익사이트-벡터"는 전체 15의 중첩을 달성하였다. "익사이트" 단독으로 보다는 300%가 증가 된다. 아직 "익사이트"의 상위 500 질의만 계산되어 졌기 때문에 "랭크덱스트" 단

〈표 2〉 익사이트-벡터 검색결과 비교

질 의	룩크-스마트	랭크덱스	익사이트	익사이트-벡터
communication	10	2	1	1
skating	10	2	1	2
virtuality	10	1	1	1
audio	5	3	0	3
video	8	2	1	1
multimedia	10	1	0	1
seoul	9	1	0	1
airlines	4	2	1	2
animal	10	2	0	1
market	3	2	1	2
Total	79	18	6	15

독보다는 더 적다. N값이 충분히 큰 값을 가진다면 보다 좋은 결과를 볼 수 있을 것이다.; 즉 보다 큰 데이터 베이스와, 보다 많은 보트, 그리고 보다 많은 객관적 결과를 볼 수 있을 것을 의미한다.

V. 결 론

결론 적으로 하이퍼링크 벡터 설정에 대한 결함과 이익적인 측면을 살 펴 볼 수 있다. 하이퍼 링크 벡터와 링크를 기반으로 한 변환된 순차 매김은 오직 링크 정보를 포함하기 때문에 등급은 문서에 나타나는 단어에 의존하지 않는다. 차라리 등급은 하이퍼링크에만 기초가 되어진다. 즉 링크가 제공하는 문서 기술과 주어진 문서에 얼마나 많은 것이 주어졌는가이다. 이것은 전통적인 등급 시스템과 공통의 몇 가지 문제를 해결한다. "키워드 검색"을 사용하는 문서는 서술과 대중적 표준을 동시에 만족할 때 높은 점수의 등급을 부여 할 수 있다. 문서의 크기는 관련성의 등급에서 요인은 보다 길지 않으며 보다 관련성이 있는 문서는 보다 짧을 수 있으나 긴 문서가 선택되어 지는 항목을 사용하지는 않는다. 이미지, 그래픽, 사운드는 기존의 재래식 방법으로는 검색되어 질 수 있으나 하이퍼링크 지시 기술에 의하여서는 검색되어질 수 있다. 이미지, 그래픽 및 기타 앵커 텍스트와 같이 서비스되어지는 경우 인덱스 엔진은 단순히 이 "꼬리-앵커" 문서의 주제로 응용되어 질 수 있는 이미지나 그래픽으로 대치할 수 있다.

하이퍼링크 벡터 설정 모형은 자동으로 어떤 특수한 영역에서 최선의 웹을 자동적으로 선택하는 응용의 하나로 유도되어질 수 있다. 동일한 문서를 지시하는 하이퍼링크의 차별적인 기술내용을 비교함으로써 새로운 개념을 도출하며 동의어를 발견할 수 있고 저장소를 설립 할 수 있다.

잠정적인 문제는 하이퍼링크 벡터 설정 기반의 엔진은 검색되어 질 수 있다. 현재 실험상에서 하이퍼링크를 중첩하여 계산하고 다중 링크로서 동일한 웹사이트에 기술을 링크한다. 링크 검색에 대한 잠정성은 장애를 유발시킬 수 있다. 즉 웹사이트에 여러번 나타난다고 하더라도 한번에 하나의 링크만 계산되어진다. 검색을 시도하는 어떤 사이트에 링크를 포함하는 다중 웹사이트를 얻는 것은 분명히 어려운 프로젝트이다.

따라서 앞으로의 작업은 전통적인 관련성 등의 위치에 하이퍼링크 벡터설정을 사용하는 것에 대한 지속적인 조사가 요구된다. 하이퍼링크 벡터 설정은 하나의 벡터설정이기 때문에 많은 질의에 관련된 문서가 아니면 결과는 랜덤해 질 수 있다. 웹과 같은 대형 하이퍼텍스트 시스템이라 할 지라도 하이퍼링크 벡터 설정은 전통적인 검색방법에 결합하여 보다 잘 동작 할 수 있다. 하이퍼링크 벡터 설정에 대한 신뢰도가 낮을 때마다, 전통적인 관련성 등급이 사용되어진다. 웹의 동적인 특성 때문에 하이퍼링크는 가끔 불량 URL를 지시하기도 한다. 그러므로 불량 링크를 검출하는 것은 실제적으로 그 다음의 단계이다. 또한 서로 다른 URL이 동일한 문서를 지시할 수도 있기 때문에 단일 문서 식별자로 그들 문서를 결합하고 중첩 URL를 검출하는 것은 지속적으로 연구할 과제이다. 마지막으로 하이퍼링크 벡터 설정은 벡터 모형으로부터 유도되어 졌으며 하이퍼링크 순서 매김이 어떻게 확률적이고 다른 검색 모형에 의하여 채용되어

질 수 있는 가에 관심이 있다.

참고 문헌

[1]. E.Selberg and O. Etzioni,"Multi-Service Search and Comparison Using the MetaCrawler," Proc. 1995 www conf., 1995; http: // draz. cs. washington.edu /papers/www4/html/Overview.html.

[2] G. R. Notes, " Internet 'Onesearch' With the Mega Search Engines," Online, Vol.20, No. 6, 1996, pp36-39.

[3] E. Keen, "Term Position Ranking: Some New Test Results,"Proc. 15th Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, 1992, pp. 66-76; http://www.acm.org/ pubs/citations/proceedings/ir/133160/p6 6-keen/.

[4] G. Singhal, A. Salton, and C. Buckley, "Length Normalization in Degarded Text, "Fifth Symp. Document Analysis and Information Retreieval, 1996; http: // www. research.att.com/~singhal/ocr-norm.ps.

[5] G. Salton, The SMART Retrieval System, Prentice-Hall, Upper Saddle River, N.J., 1971.

[6] J. Boyan, D. Freitag, and T. Joachims, " A Machine-Learning Architecture for Optiming Web Search Engine, "Proc. AAAI Workshop Internet-Based information Systems, 1996; http: //www. lb.cs.cmu.edu/afs/cs/project/reinforcement/ papers/ boyan. laser. ps

[7] R. Thompson and W.B. Croft, "Support for Browsing in an Intelligent Text Retrieval System, " Int'lJ. Man-Machine Studies, Vol. 30, No. 6, 1989, pp.639-668

[8] G. Salton, Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer, Addison Wesley Longman, Reading, Mass., 1989.

[9] W. B. Croft and H. Turtle, "A Retrieval Model for Incorporation Hypertext Links, "Hypertext 89 Proc., ACM Press, Pittsburgh, 1989, pp. 213-224.

[10] Y. Li, "Beyond Relevance Ranking: Hyperlink Vector Voting, "RIAO 97: Computer-Assisted Information Searching on Internet, 1997, McGill univer- sity, Montreal, Canada, pp.638-650.

[11] D. Dreilinger and A.E. Howe, "Experiences with Selection Search Engines Using Metasearch, "ACM Trans. Information Systems, Vol.15, No.3, July 1997, pp.195-222.