하이퍼 텍스트의 가중치 조절과 링크 구조 분석 기법을 통한 검색 엔진 성능 개선*

국민대학교 컴퓨터학부 & AITrc
이 상 호† · 강 승 식

Performance Improvement of Information Retrieval System through Weight Adjustment of Hypertext and Link Structure Analysis

Sang Ho Lee, Seung-Shik Kang
School of Computer Science, Kookmin University, Seoul, Korea

요 약

웹문서의 가장 큰 특징 중 하나는 링크 구조이다. 이 링크들을 이용하여 전체 웹문서를 커다란 하나의 네트워크로 구성할 수 있으며 이러한 네트워크를 분석함으로써 보다 중요한 문서, 보다 유용한 사이트를 찾아낼 수 있다. 전통적인 검색 모델인 벡터 모델의 성능 개선을 위해 이러한 링크 분석 기법을 활용하여 검색 정확도를 향상시키기 위한 방법을 제안한다. 또한 하이퍼 텍스트는 보다 정확한 키워드를 포함할 확률이 높으므로, 이를 가중치 계산에 적용하여 보다 정확한 결과를 산출한다.

서 론

상용화된 대부분의 검색 엔진들이 자연어 기반 질의를 처리하여 줌에도 불구하고 아직도 단일 어절로 구성된 질의가 대부분을 차지하고 있다. 단일 어절의 질의들은 그 의미가 상당히 포괄적이어서 광역 질의가 될 가능성이 아주 높다. 그러나 광역 질의에 대해서 전통적인 검색 모델들이 어느 정도 한계가 부각되고 있는 상황이며, 광역 질의들에 대한 문제를 해결하기 위해 링크 구조 분석 기법들이 지속적인 연구와 상용화가 이루어지고 있다.

단순히 링크 구조만을 사용하여 검색 엔진을 구축할 때 사용자 질의나 문서의 용어들에 대한 고려가 반영되지 않기 때문에 문제의 소지가 될 수 있다. 그래서 웹문서의 키워드 정보들을 같이 활용할 때 보다 정확도가 높은 검색엔진을 구성할 수 있다.

본 연구에서는 웹문서의 키워드 정보를 in-link text, out-link text, body text로 3단계로 구분하여 각각에 대한 별도의 가중치를 부여함으로써 보다 정확한 검색결과를 계산한다. 이와 더불어 링크 구조 분석 기법을 응용하여 키워드에 대한 가중치 정보와 함께 링크에 대한 가중치를 고려함으로써 정확도를 향상시키는 방법을 제안한다.

*본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.
†E-mail : focuschange@chol.com

기존 연구

전통적인 검색 엔진들은 주로 키워드 검색을 위주로 구성되며 본 연구에서 벡터 모델을 활용하여 하이퍼 텍스트를 분석한다. 벡터 모델은 specific query에 좋은 결과를 보여주긴 하지만 broad-topic query에는 정확도가 현저히 저하되는 경향이 있다(Brin S & Page L, 1998). 이를 보안하기 위해 링크구조 분석기법으로 활용할 것이다. 검증된 알고리즘으로는 Page Rank 알고리즘과 HITS 알고리즘등이 있다(Henzinger M, 2001). 본 연구에서는 Page Rank 알고리즘에서 in-link만을 적용하고 out-link를 사용하지 않는 방향으로 방법을 단순화시켰다.

선행 처리

보다 정확한 검색 결과를 산출해 내기 위해서 앞으로 제안

할 알고리즘 이외에 다양한 휴리스틱을 적용할 필요가 있다 (Jon M, Kleinberg, 1999). 그 몇 가지 예를 이 절에서 소개한다. 우선 중복 문서를 제거하는 문제이다. 이는 문서 내용 비교 방법을 사용할 시 프레임 구조의 문서들은 모두 제거되는 오류를 범할 수 있다. 그래서 내용보다는 문서의 링크 구조를 활용하면 보다 폭넓게 중복 제거를 할 수 있다 (Henzinger M, 2001).

링크 구조 분석 시 도메인 명을 확인하여 동일한 사이트로 연결되고 있는 링크는 모두 제외하는 것이 바람직하다. 이는 네비게이션 목적일 가능성이 크기 때문이다. 또한 한 도메인의 많은 페이지가 외부의 어떤 한 페이지로 링크되어 있을 경우 광고성 링크이거나 트래픽을 몰아줄 계약을 체결했을 가능성이 아주 높다. 이런 경우, 단일 도메인에서 특정 페이지 p로의 링크가 있는 페이지의 개수는 4~8개로 제한하는 것이 경험적으로 좋은 결과를 가져다 준다(Jon M, Kleinberg, 1999).

그 외에 다른 문제로 다음과 같은 경우를 생각해 볼 수 있다. 사용자들은 '한국 야후' 사이트에 대한 링크를 구성할 때 yahoo.co.kr 또는 www.yahoo.co.kr, kr.yahoo.com등 다양한 방법으로 구성할 수 있다. 그러나 웹 로봇은 데이터를 수집하는 단계에서 kr.yahoo.com 사이트만으로 모든 링크들을 통일시킬 것이다. 이럴 경우 링크 구조 분석 시 누락되는 링크가 발생할 수 있다. 이를 위해 링크 구조 분석 단계에서 대표 호스트를 처리할 수 있도록 하여야 한다.

제안 기법

검색 결과를 계산하는 절차로 시드 집합을 계산한 후 각 집합 원소별 가중치를 키워드 가중치와 링크 가중치로 나누어 두 가지로 부여하여 준다. 그리고 이 두 가지를 조합하여 최종 순위를 결정하게 된다. 이 방법을 anchor rank라 칭하고 이 절에서 단계별 세부 기법들을 설명한다. 우선 시드 집합을 산출할 전체 문서 집합을 3개의 개별 집합으로 구성하였다.

Fig. 1에서 화살표는 링크되어 있는 방향을 나타낸다. 문서 3을 기준으로 3개의 집합을 구분하는 예는 다음과 같다.

In-link text 집합 : 자동차 판매, 중고차.
Out-link text 집합 : 승용차, 화물차.
Body text 집합 : 종류, 승용차, 화물차, ….

위의 예와 같은 방식으로 모든 문서에 대해 3개의 집합을 구분하여 생성한다.

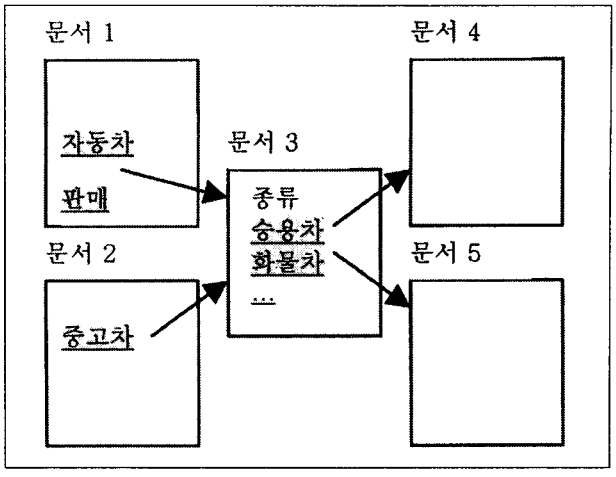

Fig. 1. 링크로 연결된 5개의 문서.

1. 시드 집합

전통적인 검색 엔진에서 사용하는 시드 집합 선정 방식과 비슷하다. 단 전체 문서를 하나의 데이터 집합으로 구성하지 않고 앞에서 설명한 예처럼 body text에 대한 집합, in-link text에 대한 집합, out-link text에 대한 집합으로 나누어 각각을 역파일로 구성한 후 이 세 집합을 통해 시드 집합을 계산한다. 시드 집합은 다음과 같이 정의한다.

I(q) : 질의 q에 대한 In-link text 시드 집합.
O(q) : 질의 q에 대한 out-link text 시드 집합.
B(q) : 질의 q에 대한 body text 시드집합.

이 때 질의 q에 대한 시드 집합 S(q)는 다음과 같다.

$$S(q) = I(q) + B(q) - O(q) \quad \cdots (1)$$

위 정의에서 O(q)를 직접 삭제하게 되면 실제로 아주 연관성이 높은 문서도 out-link가 존재한다는 이유만으로 완전히 배제될 수가 있다. 이런 경우를 막기 위해 전체 시드 집합을 계산할 때 단순히 out-link text의 값들을 무조건 삭제하지 않고 각 집합별 벡터 내적으로 계산된 가중치를 적용하여 가중치에 대한 연산을 통해 특정 임계값을 정한 후 배제하도록 하였다. 이럴 경우 질의에 맞지 않는 내용이면서 out-degree가 높은 문서(Hub 사이트와 같이)들은 자동으로 가중치 값이 하향되므로 시드 집합에서 배제될 수가 있다.

2. 가중치 부여 방법

우선적으로 고려할 부분은 용어 가중치와 링크 가중치를 서로 별개로 계산한다는 것이며, 더더욱 중요하게 고려할 점은 in-link text와 out-link text에 대해 서로 다른 가중치

를 부여한다는 것이다. 링크를 생성하는 사람은 링크 텍스트의 문구를 링크되어 있는 문서에 대해 보다 정확한 설명이나 주제로 기술할 경우가 많다. 역으로 생각하면 문서내의 링크는 문서 자신의 내용 보다는 링크되어 있는 문서에 대한 정보를 수록할 경우가 많다. 그러므로 in-link text에 대한 가중치는 문서의 내용이나 out-link text보다 오히려 높은 가중치를 부여하면 좋은 결과를 얻을 수 있다. 시드 집합 S(q)에서 j번째 문서 d_j에 대한 i번째 용어 t_i의 가중치는 다음과 같이 계산한다.

$$w(t_i,d_j)=\alpha\times W_I+\beta\times W_B-\gamma\times W_O$$
$$W_I=\text{집합}I(q)\text{에서문서}d_j\text{에대한용어}t_i\text{의가중치}$$
$$W_B=\text{집합}B(q)\text{에서문서}d_j\text{에대한용어}t_i\text{의가중치} \quad \cdots (2)$$
$$W_O=\text{집합}O(q)\text{에서문서}d_j\text{에대한용어}t_i\text{의가중치}$$
$$\alpha+\beta+\gamma=1$$

용어 가중치 W_I, W_B, W_O는 벡터모델에서 일반적으로 사용하고 있는 tf-idf 방식을 사용하여 각각을 동일한 방법으로 계산한다(김영철 등, 2001). 아래 수식은 tf-idf 방식에 대한 설명이다.

$$w_{i,j}=tf_{i,j}\times\log\frac{N}{n_i}$$
$$w_{i,j}=j\text{번째문서에대한}i\text{번째용어의가중치}$$
$$tf_{i,j}=j\text{번째문서}i\text{번째용어의빈도수} \quad \cdots (3)$$
$$N=\text{전체집합에대한문서수}$$
$$n_i=N\text{에서문서}d_i\text{가출현한빈도수}$$

수식(2)에서 α, β, γ는 in-link text, out-link text, 본문에 대한 가중치 비율을 조율하기 위해 사용하는 상수값이다. 이 실험에서는 각각 0.67, 0.23, 0.1의 값을 사용하였다. 이 값은 반복적인 실험 끝에 얻어낸 값이긴 하나 실제적으로 좀더 세밀히 조절해 볼 필요는 있다. α의 값을 높일수록 in-link text의 내용에 비중을 많이 두는 것이므로 이 실험에서 in-link text를 얼마나 비중있게 적용하였는지 알 수 있다. 만일 α,γ의 값을 0으로 설정한다면 하이퍼텍스트를 고려하지 않은 일반적인 벡터 모델과 동일한 가중치를 산출하게 될 것이다. 그리고 γ의 값을 높게 준다면 재현율을 감소시켜 정확도를 조금 향상시킬 수는 있으나 지나치게 많이 주게 되면 오히려 정확도가 감소될 수 있다. 이는 out-link text 집합을 다른 두 집합과 완전히 구분하여 생성하였기 때문에 발생하는 현상으로 항상 전체 가중치가 0보다 작은 값을 가지는 경우가 발생할 수 있다. 이런 경우를 피할 수 있도록 적절히 γ값을 낮추어야 한다.

문서 d에 대한 링크 가중치는 in-link degree를 직접 반영하였다. Out-link degree를 고려하여 반영하게 되면 좀 더 정확도를 높일 수 있으나 이런 방법만으로도 충분히 원하는 정확도를 얻을 수 있었다. 다만 특이하게 많은 degree가 존재하는 사이트에 대한 편차를 줄이기 위해 어느 임계값으로 degree를 제한할 필요가 있다. 이 실험에서는 100정도로 제한하여 사용하였다.

3. 랭킹 부여 방법

전체 문서의 순위는 용어의 가중치가 높을수록, 그리고 링크 가중치의 값이 높을수록 높은 우선 순위를 부여하게 된다. 두 종류의 가중치 값이 서로 다른 편차를 보이는데 이는 최종 결과에 나쁜 영향을 줄 수 있다. 그래서 두 값을 다음 수식들로 정규화 하여준다. 다음은 용어 가중치에 대한 정규화 식이다.

$$Norm(w_t)=1-\frac{1}{\sqrt{w_t}} \quad \cdots (4)$$

수식(4)에서 w_t는 임의의 문서에 대한 용어의 가중치로 수식(2)의 $w(t_i, d_j)$값을 나타낸다. 다음은 링크 가중치에 대한 정규화 식이다.

$$Norm(l_d)=(1-d)+d\times\left(1-\frac{1}{\sqrt{l_d}}\right) \quad \cdots (5)$$

수식(5)에서 l_d는 문서 d의 링크 가중치를 나타낸다. 이 실험에서 링크 가중치로 in-link degree를 사용하였기 때문에 여기서는 in-link degree의 값이 l_d의 값이 된다. 의 값은 damping factor의 역할을 하게 된다(Henzinger M, 2001). 이 실험에서는 이 값을 0.8로 적용하였을 때 좋은 결과를 얻을 수 있었다.

두 정규화된 가중치에 대한 값을 고려한 최종 질의-문서 유사도는 용어 가중치와 링크 가중치를 2차원 좌표상의 점

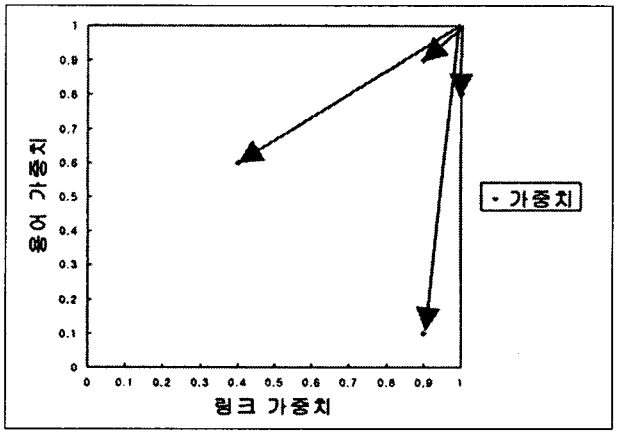

Fig. 2. 유사도 계산 방법.

(1,1)과 가까울 수록 높은 순위를 부여한다. Fig. 2는 이러한 방법을 보여 준다. Fig. 2에서 벡터의 크기(화살표)가 실제 유사도 값으로 결정된다. 다음은 유사도를 계산식이다.

$$Sim(q,d) = 1 - \sqrt{Norm(w_t)^2 + Norm(l_d)^2} \quad \cdots (6)$$

실험 및 평가

이 실험에서 2,300만개 정도의 웹문서를 대상으로 하였다. 무작위의 웹문서들이기 때문에 R-Precision 방식과 Average-Precision방식으로 정확도를 측정하였으며 문서의 재현율에 대해서는 고려하지 않았다.

테스트 질의는 대형 검색 포탈 사이트의 상위 질의 20여개를 참조하여 기존의 벡터 모델과 이 논문에서 제안한 모델과 비교 분석하였다. R값은 10으로 했으며, 검색 질의의 가장 큰 부분을 차지하고 있는 broad topic query에 대한 결과를 측정한 것이다.

Table 1. Vector model과 Anchor rank 모델의 정확도 측정

Cutoff	Vector Model		Anchor Rank	
	Avg	Precision	Avg	Precision
1	0.5000	0.5000	1.0000	1.0000
2	0.5000	0.3333	1.0000	1.0000
3	0.6030	0.5550	1.0000	0.8877
4	0.6342	0.5000	0.9860	0.7917
5	0.6400	0.4333	0.9505	0.8000
6	0.6400	0.3583	0.9348	0.7477
7	0.5933	0.3540	0.9160	0.7378
8	0.5725	0.3317	0.8913	0.7500
9	0.5725	0.2943	0.8880	0.7223
10	0.5417	0.3222	0.8787	0.6833
Avg	0.5798	0.3982	0.9445	0.8121

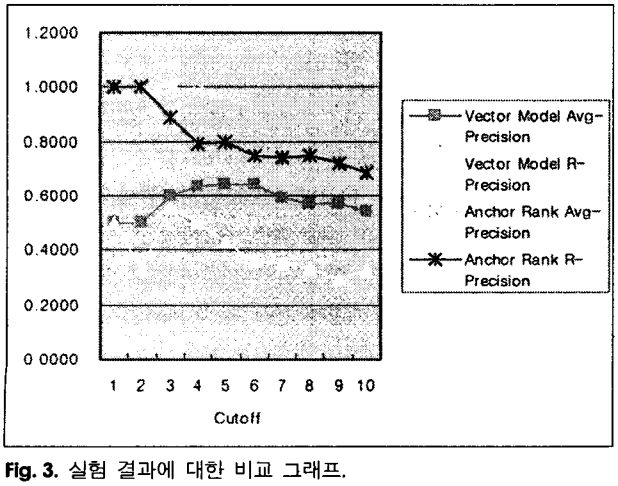

Fig. 3. 실험 결과에 대한 비교 그래프.

Table 1을 보면 전통적인 벡터 모델을 사용하였을 때 보다 이 논문에서 제안한 Anchor Rank를 사용하였을 경우 평균 두배 정도 정확도가 증가됨을 볼 수 있다. Fig. 1은 위의 결과를 도식화한 것이다. Anchor Rank 모델의 경우 cutoff 값이 올라갈수록 정확도가 점차 감소하는 것을 볼 수 있다. 그러나 벡터 모델의 경우는 그렇지 않다. 이 의미는 Anchor Rank 모델의 경우 정답 문서가 상위로 잘 랭크 되는 것을 나타낸다. 벡터 모델일 경우 전체적으로는 차츰 감소하나 원하는 문서가 상위에 잘 랭크 된다고 볼 수 없다. 이러한 차이는 cutoff값을 늘리거나 실험 횟수를 늘려보면 줄어드는 경향이 나타난다. 그러나 전반적인 검색 엔진의 성능 면에서 볼 때 정확한 문서가 상위에 랭크 되는 것이 바람직하기 때문에 벡터 모델보다 이 논문에서 제안하는 방법이 훨씬 효율적이라는 것을 알 수 있다. 참고로 벡터 모델의 결과가 다른 연구와 차이가 보이는 것은 형태소 분석을 통한 키워드 추출 방법 및 용어 처리 부분의 문제이며 이 연구를 하는데 있어서는 중요한 요인은 아니다.

결론 및 향후 과제

테스트 컬렉션이 얼마나 잘 정제되어 있는냐는 정확도에 큰 영향을 주고 있다. 이 실험에서 사용된 컬렉션은 공인된 컬렉션을 사용하지 않고 웹문서를 임의로 수집하여 실험한 것이기 때문에 잘 정제되어 있지 않은 상태이다. 또한 대표 호스트를 반영하지 않았고, 미러 사이트와 같은 구조상의 중복되는 문서들을 제대로 처리되지 않았다. 그리고 웹로봇이 수집하지 못한 문서들에 대한 링크들을 모두 없는 문서로 간주하였기 때문에 생각보다 많은 문서들이 배제되어 있다. 이러한 부분은 링크 구조 분석 단계에서 후처리로 한번 더 수집하여 주는 것이 바람직 할 것이다.

키워드 가중치는 tf-idf 방식을 골격으로 작성되었으며 in-link text, out-link text, 문서 본문의 세가지 집합으로 나누어 별도 계산을 수행했다. 이 부분에서 세 집합을 하나의 집합으로 보고 in-link text와 out-link text에 대한 가중치 비율을 조율하는 방법으로 실험해 볼 가치가 있다. 그리고 단순한 tf-idf 방식보다 좀더 발전된 모델을 사용한다면 좀 더 정확해 질 수 있을 것이다.

링크 가중치를 계산하는데 있어서 단순한 degree만을 반영하는 것은 치명적인 오류를 발생시킬 수 있다. 이는 특정 페이지를 가리키는 모든 페이지들에 대한 out-link degree까지 반영될 때 문서에 대한 링크 구조를 더욱 명확하게 표현할 수 있다. Page Rank 알고리즘이나 HITS 알고리즘과 같이 링크 구조에 대해 좀더 세부적으로 반영한다면 현재

실험 결과보다 월등히 좋은 결과를 보여줄 것이다.

REFERENCES

Brin S, Page L(1998) : *"The anatomy of a large-scale hypertextual web search engine", Proceedings of the 7th International World Wide Web Conference, pp107-117*

Jon M, Kleinberg(1999) : *"Authoritative sources in a hyper linked environmemt", Journal of the ACM, vol 46, no 5, pp604-632*

김영철 등(2001) : 최신 정보검색론, 홍릉과학출판사, *pp31-82*

Henzinger M(2001) : *"Web Information Retrieval", Google*