# Anchor Text 정보와 링크 정보를 이용한 정보 검색 모델

한기덕O* 정성원* 허희근* 이교운** 권혁철*
*부산대학교 정보컴퓨터 공학부
**울산과학대학교 컴퓨터정보학부
templer@pusan.ac.kr swjung@pusan.ac.kr riniya@pusan.ac.kr kwlee@mail.uc.ac.kr hckwon@pusan.ac.kr

# Information Retrieval Model Using Anchor Text Information and Link Information

Gi-deok HanO* Sung-won Jung* Hee-keun Heo* Kyo-woon Lee** Hyuk-chul Kwon*
*Dept. of Computer Science and Engineering, Pusan National University
**Dept. of Computer and Information, Ulsan College

**요 약**

90년대 이전에 정보 검색에 대한 연구는 문서의 내용을 기반으로 한 연구가 주류였으며, 90년대에는 링크를 이용한 연구가 활발하였다. 90년대 말에 Page Rank와 HITS가 링크를 이용한 연구의 대표적 사례이며, 최근에는 문서의 내용과 링크 정보를 같이 이용하는 연구가 많이 발표되고 있다. 본 논문도 문서의 정보와 링크 정보를 이용한 새로운 검색 모델을 제시하고자 한다. 본 논문에서 사용하는 링크 정보는 수집된 문서에서 추출한 Page Rank의 가중치와 한 페이지를 가리키는 링크들의 목록이며, 사용하고자 하는 문서의 정보는 본문 내용과 Anchor Text이다. 링크 정보와 문서 정보를 이용하여 Anchor 벡터와 문서 벡터를 만들고, 각각 질의어 벡터와 Cosine Measure를 하여 값을 구한 후, 더한 값을 해당 문서의 가중치로 하여 검색에 이용한다.

## 1. 서 론

정보 검색은 90년대 이전에는 단어 출현 빈도 등과 같은 해당 문서의 내용을 기반으로 문서를 검색하는 방법이 주류를 이루었으며, 90년대에는 링크 정보를 이용한 수많은 연구가 진행되었다. 이런 링크를 이용한 연구는 90년대 후반에 접어들어 HITS[13]와 PageRank[12]가 링크를 이용한 대표적인 사례로 인정되면서, 이후 링크의 정보와 문서의 정보를 혼합하는 연구[2][11][14], 링크의 정보와 다른 모델을 결합하는 연구[7][8] 등이 발표되었고, 최근까지 그 추세는 계속되고 있다. 본 논문은 최근 추세와 같이 링크의 정보와 문서의 정보를 결합하는 새로운 모델을 제시하고자 한다. 논문에서 사용하는 링크 정보는 수집된 문서에서 추출한 Page Rank의 가중치와 한 페이지를 가리키는 링크들의 목록이며, 사용하고자 하는 문서의 정보는 본문 Text와 Anchor Text이다. 본 논문은 Page Rank의 가중치, Anchor Text와 한 페이지를 가리키는 링크의 목록을 이용하여 TF*IDF와 유사하게 적용하여 Anchor 벡터를 만드는 모델을 제시한다. 제안한 모델을 사용하여 만들어진 Anchor 벡터와 일반적인 기법인 TF*IDF로 만들어진 문서 벡터에 각각 질의어 벡터와 Cosine Measure를 적용한다. Cosine Measure의 적용 결과로 구해진 2개의 값을 더한 값을 해당 문서의 가중치로 하여서 검색을 하는데 이용한다.

## 2. 관련 연구

웹 문서에서 링크는 웹 문서 간을 연결하는 역할을 수행하므로 웹 관련 연구에 있어서 링크는 매우 중요한 요소이다. 링크 정보는 다양하게 이용될 수 있는데, 그 한 예로써, Yitong Wang과 Masaru Kitsuregawa[1]는 링크를 이용한 클러스터링을 제안하였다. 이 기법은 링크의 정보 중에 하나인 Co-citations, Coupling, Hub, Authority 등의 정보를 이용하여 클러스터링에 응용한다.

Rong Jin과 Susan Dumais[2]는 링크의 정보와 Content 정보를 혼합하는 알고리즘을 제안하였다. 간략하게 설명하자면, 질의어와 문서의 유사도, 링크로 연결된 문서와 문서 간의 유사도, 링크 정보를 가진 문서의 중요성(Page Rank, Hub, Authority 점수) 등을 이용하여 Content based 점수와 링크-based 점수를 혼합하여 combined 점수를 구하는 알고리즘을 제시하였다.

Brian D. Davison[5]는 Anchor Text가 검색의 성능의 향상에 도움을 줄 수 있다는 사실을 보여주었고, Anchor Text의 성능과 관련한 실험[9]이 최근에 발표되기도 하였다. 실제로 Anchor Text를 이용한 사례로 Nick Craswell, David Hawking와 Stephen Robertson[3]는 Anchor Text에서 출현한 단어를 이용하여 사이트 내의 연관 문서를 찾는 방법을 제시하였다.

이 외에도 W. Kraaij, T. Westerveld와 D. Hiemstra[10]는 Content와 Anchor에 대한 Language Model를 적용하는 새로운 방법을 제시하기도 하였다.

한국에서 이루어진 연구로는 "용어가중치 결합이 검색 효율성에 미치는 영향 연구"[6]를 들 수 있는데, 이 논문은 용어가중치 결합이 어느 정도의 성능을 낼 수 있는지를 보여주었으며, 이 결과를 통해 용어(Term)만을 이용한 검색 시스템의 성능을 알 수 있다. 그 외에 최근 연구로써 "하이퍼 텍스트의 가중치 조절과 링크 구조 분석 기법을 통한 검색 엔진 성능 개선"[4]이란 논문에서는 In-링크와 Out-링크의 Anchor Text를 이용한 검색 방법을 제시하고 있다.

## 3. 제안하는 방법

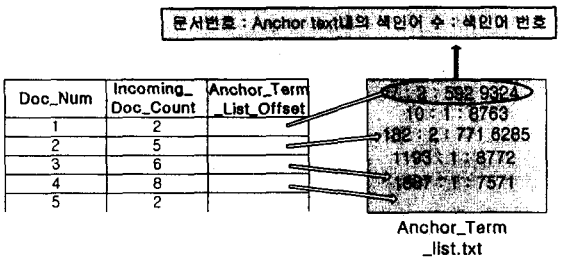
본 논문은 Page Rank의 가중치, 한 페이지를 가리키는 모든 링크의 Anchor Text를 가지고 만들어진 Anchor 벡터와 일반적인 TF*IDF 검색에서 이용되는 질의어 벡터와 문서 벡터를 가지고 Cosine Measure를 적용해서 가중치를 구하고, 구해진 가중치를 가지고 검색을 하는데 이용한다.

### 3.1 한 문서를 가리키는 링크들의 목록 만들기

문서를 수집하는 작업은 부산대학교 인공지능 연구실에서 구현한 문서 수집기(Web Crawler)를 이용하여 이루어졌으며, 수집한 문서를 연구실에서 구현한 색인기를 통해 색인 한 후, 한 문서를 가리키는 링크들의 목록을 만들게 된다.

사용될 Page Rank의 가중치는 자체적으로 수집된 문서만을 가지고 Page Rank 알고리즘을 적용한 결과를 이용한다.

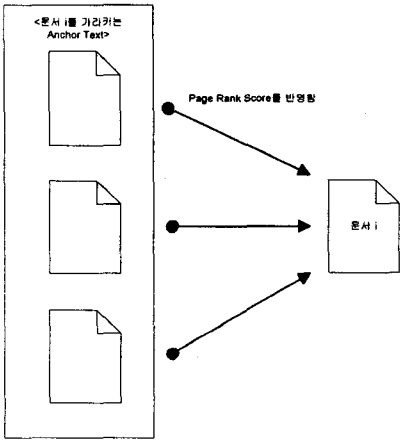
그림 3-1은 링크들의 목록을 관리하는 사전의 구조이다.




[그림 3-1] Anchor Text와 링크의 정보를 관리하는 사전의 구조도

### 3.2 링크들의 목록을 이용하여 Anchor 벡터 만들기

그림 3-2은 Anchor 벡터를 만드는 방법을 간단하게 표현한 그림이다.




[그림 3-2] 문서 i의 Anchor 벡터를 만드는 방법

Anchor 벡터는 다음과 같은 방법으로 만들어진다.

- ▷ 한 문서를 가리키는 모든 Anchor Text를 합하여 하나의 Anchor Text 집합을 만든다.
- ▷ 벡터는 TF * IDF 방식을 이용하여 구한 값들의 집합이다. (IDF는 IAF(Inverse Anchor Frequency)로 바꾼다.)
- ▷ Anchor 벡터의 TF는 링크에서 해당 단어가 출현할 때마다 정규화 처리(사용하는 정규화방법은 최대-최소정규화이다)를 한 Page Rank 가중치를 더하는 방식으로 계산한다. 이 계산 방식은 벡터의 구축시 각 링크의 가중치를 Page Rank 가중치를 이용하여 중요한 링크에 더 많은 가중치를 주게 한다.
- ▷ Anchor 벡터의 IAF는 해당 단어가 출현한 Anchor Text 집합의 수를 이용하여 구한다.
- ▷ 한 Host에서 문서 i로의 링크가 많아 단어의 빈도가 증가하는 문제를 막기 위해 제한 값을 둔다. ( 본 논문에서는 10으로 제한 값을 두었다. )
- ▷ 한 Anchor Text에서 출현한 단어의 빈도는 출현했으면 1, 없으면 0으로 둔다. ( 본 논문에서는 여러 링크의 Anchor Text에서 동시 출현한 단어의 빈도가 가치가 있다고 간주하여 실험하였다. )

위에서 언급한 처리의 수식은 다음과 같다

**Page Rank의 값을 위한 정규화 수식 :**

$$npr_i = \lambda \times \frac{pr_i - Min_{pr}}{Max_{pr} - Min_{pr}} + 1$$

$npr_i$ = 정규화 처리를 한 i번째 문서의 Page Rank의 값
$pr_i$ = i번째 문서의 Page Rank의 값
$Min_{pr}$ = Page Rank의 값 중 최소 값
$Max_{pr}$ = Page Rank의 값 중 최대 값
$\lambda$ : 임의의 값 ( 실험에서는 4를 사용함 )

**tf - iaf 수식 :**
(tf는 다른 처리 없이, 최대값 제한만 255로 두었다.)

$$Score(a_i,t_j) = tf_{ij} \cdot \log \frac{(N+1)}{n_j}$$

$a_i$ = i번째 Anchor Text 집합
$t_j$ = j번째 단어
$tf_{ij}$ = i번째 Anchor Text 집합에서 j번째 단어가 출현한 빈도
N = 전체 Anchor Text 집합 수
$n_j$ = j번째 단어가 출현한 Anchor Text 집합의 총 수

### 3.3 질의어, Anchor, 문서 벡터를 이용하여 해당 문서의 가중치 계산하기

벡터간의 Cosine Measure 연산을 통하여 해당 문서의 가중치를 계산하는 수식은 다음과 같다.

$$Score(d_i) = \lambda \times f(Q,A_i) + (1-\lambda) \times f(Q,D_i)$$

$d_i$ : i번째 문서
$\lambda$ : 1보다 작은 임의의 값
Q : 질의어 벡터
$A_i$ : i번째 문서의 Anchor 벡터
$D_i$ : i번째 문서의 문서 벡터
f(x,y) : x와 y의 cosine measure를 구하는 함수

### 4. 실험 및 결과

연구실 내에서 구축한 1000만 건 문서를 가지고 실험을 하였으며, 실험을 위한 데이터 셋(Data Sets)의 특성은 다음과 같다

표 4-1에서 단어는 중복을 제거하지 않은 상태이다.

| 속성 \ 문서 집합 | 1~50만 | 50만 1~500만 | 500만 1~1154만 8800 | 전체 문서 |
|---|---|---|---|---|
| 링크의 수 | 5578만 6504 | 5538만 3576 | 1823만 6956 | 1억2940만7036 |
| 한 Anchor에서 출현한 단어 최대 수 | 694 | 710 | 692 | 710 |
| 한 Anchor에서 출현한 단어 평균 수 | 1.81 | 1.92 | 2.29 | 1.92 |
| Anchor Text 집합에 출현한 단어 최대 수 | 999만 5964 | 111만 4632 | 12만 3067 | 999만 5964 |
| Anchor Text 집합에 출현한 단어 평균 수 | 201.83 | 23.65 | 6.37 | 21.57 |
| 출현한 모든 단어 수 | 1억 91만 6634 | 1억641만 7869 | 4174만 5244 | 2억4907만9747 |

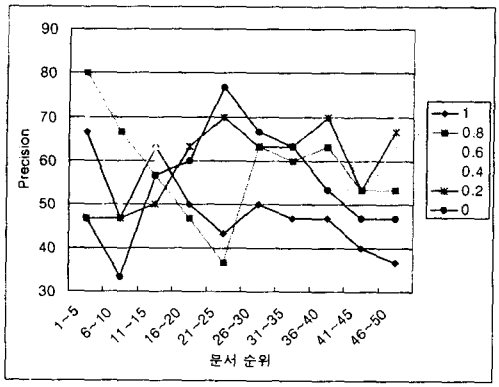
[표 4-1] Anchor Text의 정보 1

| 속성 \ 값 | 수치 값 |
|---|---|
| 단어가 존재하는 Anchor Text 집합 수 | 461만 9940 |
| 중복된 단어의 제거 후, 한 Anchor Text 집합에서 출현한 단어 최대 수 | 4만 4142 |
| 중복된 단어의 제거 후, 한 Anchor Text 집합에서 출현한 단어 평균 수 | 2.39 |
| 중복 제거 후, 단어가 존재하는 Anchor Text 집합들에서 출현한 단어 평균 수 | 5.97 |
| 중복된 단어의 제거 후, 출현한 단어의 총 수 | 177만 4388 (본문에서의 총 단어 : 2467만 4444개 ) |

[표 4-2] Anchor Text의 정보 2

실험에 사용한 질의어는 대형 검색 포탈 사이트의 인기 검색

단어들을 참조하였으며, 본 논문에서 제안한 모델의 성능 평가는 검색 결과 중 상위 50건에 대한 정확도(Precision)를 이용하여 하였으며, $d_i$의 가중치를 계산하는 수식에서 λ의 값을 다양하게 변환하여 실험을 해 보았다. λ가 0.0인 경우의 결과는 기존의 tf-idf 모델의 성능이 된다.




[그림 4-1] 수치값에 따른 성능을 보여주는 그래프

| 문서 순위 | 정확도 | 표준편차 |
|---|---|---|
| 1~5 | 66.67 | 1.86 |
| 6~10 | 46.67 | 1.97 |
| 11~15 | 63.33 | 1.83 |
| .. | .. | .. |
| 41~45 | 40.00 | 1.67 |
| 46~50 | 36.67 | 1.60 |
| 합계 | 49.00 | 16.08 |

[표 4-3] λ=1.0의 결과

| 문서 순위 | 정확도 | 표준편차 |
|---|---|---|
| 1~5 | 80.00 | 0.89 |
| 6~10 | 66.67 | 1.50 |
| 11~15 | 56.67 | 1.17 |
| .. | .. | .. |
| 41~45 | 53.33 | 2.34 |
| 46~50 | 53.33 | 1.21 |
| 합계 | 58.00 | 7.29 |

[표 4-4] λ=0.8의 결과

| 문서 순위 | 정확도 | 표준편차 |
|---|---|---|
| 1~5 | 70.00 | 1.05 |
| 6~10 | 50.00 | 1.64 |
| 11~15 | 63.33 | 1.72 |
| .. | .. | .. |
| 41~45 | 56.67 | 1.72 |
| 46~50 | 70.00 | 1.22 |
| 합계 | 62.67 | 7.89 |

[표 4-5] λ=0.6의 결과

| 문서 순위 | 정확도 | 표준편차 |
|---|---|---|
| 1~5 | 53.33 | 1.37 |
| 6~10 | 46.67 | 1.50 |
| 11~15 | 60.00 | 1.67 |
| .. | .. | .. |
| 41~45 | 53.33 | 1.37 |
| 46~50 | 63.33 | 1.33 |
| 합계 | 60.00 | 7.97 |

[표 4-6] λ=0.4의 결과

| 문서 순위 | 정확도 | 표준편차 |
|---|---|---|
| 1~5 | 46.67 | 1.51 |
| 6~10 | 46.67 | 1.86 |
| 11~15 | 50.00 | 1.52 |
| .. | .. | .. |
| 41~45 | 53.33 | 1.86 |
| 46~50 | 66.67 | 1.51 |
| 합계 | 59.33 | 8.91 |

[표 4-7] λ=0.2의 결과

| 문서 순위 | 정확도 | 표준편차 |
|---|---|---|
| 1~5 | 46.67 | 1.63 |
| 6~10 | 33.33 | 1.03 |
| 11~15 | 56.67 | 1.17 |
| .. | .. | .. |
| 41~45 | 46.67 | 1.63 |
| 46~50 | 46.67 | 1.37 |
| 합계 | 55.00 | 7.74 |

[표 4-8] λ=0.0의 결과

상위 50건의 정확도는 λ가 0.6일 때, 가장 좋으며, 상위 10건의 정확도는 λ가 0.8일 때가 가장 높다. 표 4-3에서와 같이 λ를 1.0(Anchor 벡터만 이용했을 때)으로 했을 때의 결과를 보면, 상위 10건의 정확도는 좋으나, 상위 50건의 정확도는 낮은 것을 볼 수 있다. 또한, 다른 수치값의 결론에 비해 합계의 표준편차가 심하다. 이러한 결과가 나온 이유는 Anchor 벡터만으로 검색을 했을 때, Anchor Text가 문서의 내용을 대표하지 못하거나 혹은 Anchor Text에 나타나지 않은 질의어로 검색을 한다면 성능이 나빠지기 때문이다. 실제로 표 4-2를 보면, 본문에서 색인된 단어 수는 2467만 4444개인데 비해, Anchor Text에서 색인된 단어 수는 177만 4388개로 적은 것을 알 수 있다.

결론을 내린다면, Anchor Text의 단어에 가중치를 주게되면 정보 가치가 높은 페이지가 높은 순위에 나타나는 반면 Anchor Text가 포용할 수 있는 단어의 수가 적음에 따라 다양한 질의어를 사용하여 검색을 하였을 때, 어떤 질의어에 대해서는 매우 낮은 성능을 나타내게 된다. 이것에 대한 대안으로 검색한 문서의 본문의 내용을 사용할 수 있다. 문서의 본문의 내용은 Anchor Text에 비해 정보 가치가 높은 페이지를 높은 순위로 올리는 성향이 낮으나, 안정적인 성능을 유지하는 것을 위의 6개의 표를 비교해 보면 알 수 있다.

## 5. 결론 및 향후 과제

Anchor Text가 검색에 유용한 정보라는 것은 이미 많은 연구에 의해 인정되고 있다. 이 논문은 이 Anchor Text를 링크 정보와 함께 정제 및 처리를 하여 더욱 유용하게 사용할 수 있는 모델을 제시하였다. 실험을 하면서 수식의 수치 값이나 변환 등이 있었으나, 다양한 수식이나 알고리즘을 적용하여 실험하지 않았고, 사용한 데이터 셋이 공인된 콜렉션이 아니므로, 실험의 결과가 제안한 모델의 성능을 정확하게 보여준다고는 할 수는 없다. 그러나 Anchor Text와 링크를 분석하여 얻어낸 정보는 Anchor Text가 실제 웹 문서에서 어떤 식으로 존재하는지 보여주었으며, 이런 분석을 통해 현재 웹 문서 안의 Anchor Text의 분포 및 형태에 보다 적합한 수식 및 알고리즘을 알아낼 수 있을 것이다. 그래서 향후 과제로써 이번 연구를 바탕으로 Anchor Text를 처리하는 적합한 수식 및 알고리즘을 찾는 방법을 다양한 각도로 연구할 계획이다.

## 참고문헌

[1] Yitong Wang, and Masaru Kitsuregawa. Link Based Clustering of Web Search Results. Second International Conference on Advances in Web - Age Information Management (WAIM), 2001.
[2] Rong Jin, and Susan Dumais. Probabilistic combination of content and links. Proc. of ACM SIGIR '01, pages 402 - 403, 2001.
[3] Nick Craswell, David Hawking and Stephen Robertson. Effective site finding using link anchor information. Proc. of ACM SIGIR '01, pages 250 - 257, 2001.
[4] 이상호, 강승식. 하이퍼 텍스트의 가중치 조절과 링크 구조 분석 기법을 통한 검색 엔진 성능 개선. 제 15회 한글 및 한국어 정보처리 학술대회, 2003.
[5] Brian D. Davison. Topical locality in the web. Proc. of ACM SIGIR '00, pages 272 - 279, 2000.
[6] 최성환, 정영미. 용어가중치 결합이 검색 효율성에 미치는 영향 연구. 한국정보과학회 봄 학술발표논문집 Vol. 29. No. 1, pages 481 - 483, 2002.
[7] I. Silva, B. Riberiro-Neto, P. Calado, N. Ziviani. Link-based and content-based evidential information in a belief network model. Proc. of ACM SIGIR '00. pages 96 - 103, 2000.
[8] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. Proc. of ACM SIGIR '98. pages 104 - 111, 1998.
[9] Nadav Eiron, Kevin S. McCurley. Analysis of Anchor Text for Web Search. Finded in CiteSeer, 2003.
[10] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In Proc. of the 25th annual international ACM SIGIR conference on research and development in information retrieval, pages 27 - 34. Association for Computing Machinery, 2002
[11] Ronald Fagin, Ravi Kumar, Kevin S. McCurley, Jasmine Novak, D. Sivakumar, John A. Tomlin, and David P. Williamson. Searching the workplace web. In Proceedings of the Twelfth International World Wide Web Conference, Budapest, 2003.
[12] S.Brin and L.Page. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1-7):107-117, 1998.
[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. Proc. of ACM SIGIR '98. pages 668 - 677, 1998.
[14] Thijs Westerveld, Wessel Kraaij, and Djoerd Hiemstra. Retrieving web pages using content, links, URLs and anchors. In Proc. 10th TREC, pages 663-672, 2001.