구조적 정보와 색인어 정보를 결합한 검색 모델 개발

임성신° 한기덕 권혁철
부산대학교 정보컴퓨터 공학부
sslim@pusan.ac.kr templer@pusan.ac.kr hckwon@pusan.ac.kr

Development of Retrieval Model Using Structure Information and Term Information

Sung-sin Lim° Gi-deok Han Hyuk-chul Kwon
Dept. of Computer Science and Engineering, Pusan National University

요 약

인터넷 정보의 축적량이 증가함으로 인해 사용자는 원하는 정보를 찾기가 더욱 어려워졌다. 따라서 수많은 문서들 중에서 원하는 정보를 효과적으로 찾아주는 정보검색 시스템의 중요성이 증가하게 되었으며 이에 대한 연구도 활발히 진행되었다. 인터넷 문서에서 추출할 수 있는 정보들은 링크 정보, Anchor Text 정보, Title Text 정보, 본문 Text 정보 등이 있으며, 이런 정보들을 이용한 수많은 정보검색 시스템이 개발되거나 모델이 연구되고 있다. 본 논문에서는 기존에 이용되어 왔던 일반적인 추출 정보들을 정제 및 처리를 통해 성능을 높일 수 있는 방안을 연구했던 선행 연구를 기반으로 한 실험 결과 및 사이트 가중치를 추가한 모델을 제시한다.

1. 서 론

전 세계의 호스트 컴퓨터가 2000년 1억 개에서 2003년 2억 3,000만개로 늘었고 우리나라의 인터넷 이용자수도 1,900만에서 2,900만으로 증가하였다. 이와 같은 인터넷 인프라의 증가로 인터넷의 발전 및 효율성이 증가하였으나, 인터넷 정보의 축적량이 증가함으로 인해 사용자는 원하는 정보를 찾기가 더욱 어려워졌다. 따라서 수많은 문서들 중에서 원하는 정보를 효과적으로 찾아주는 정보검색 시스템의 중요성이 증가하게 되었으며 이에 대한 연구도 활발히 진행되었다. 인터넷 문서에서 추출할 수 있는 정보들은 링크 정보, Anchor Text 정보, Title Text 정보, 본문 Text 정보 등이 있으며, 이런 정보들을 이용하여 수많은 정보검색 시스템이 개발되거나 모델이 연구되어왔다. 예를 들어 HITS[13]와 PageRank[12]는 링크를 이용한 대표적인 모델이며, 링크의 정보와 문서의 정보를 혼합하는 연구[2][11][14], 링크의 정보와 다른 모델을 결합하는 연구[7][8] 등 다양한 연구가 진행되어왔다. 선행 연구로써 Anchor Text 집합 Vector를 구축하는 방법[15]과 링크 정보, Title Text 정보를 추가하여 가중치 실험[16]을 했으며, 본 논문에서는 기존에 이용되어 왔던 일반적인 추출 정보들을 정제 및 처리를 통해 성능을 높일 수 있는 방안을 연구했던 선행 연구를 기반으로 한 실험 결과 및 사이트 가중치를 추가한 모델을 제시한다.

2. 관련 연구

선행 연구로써 "Anchor Text 정보와 링크 정보를 이용한 정보 검색 모델"[15]에서 Page Rank의 가중치, Anchor Text와 한 문서를 가리키는 링크의 목록을 이용하여 TF*IDF와 유사하게 적용하여 Anchor Text 벡터를 만드는 방법과 모델을 제시하였으며, "한국어 웹 문서의 구조적 정보와 색인어 정보를 결합한 검색 성능 향상"[16]에서 이전 연구의 모델에 Title Text 정보와 링크 정보를 추가하여 가중치 조절에 따른 검색 성능의 평가 실험을 하였다.

"Link Based Clustering of Web Search Results"[1]에서는 링크를 이용한 클러스터링을 제안하였다. 이 기법은 링크의 정보 중에 하나인 Co-citations, Coupling, Hub, Authority 등의 정보를 이용하여 클러스터링에 응용했으며, "Probabilistic combination of content and links"[2]에서는 링크의 정보와 Content 정보를 혼합하는 알고리즘을 제안하였다.

"Topical locality in the web"[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 "Effective site finding using link anchor information"[3]에서는 Anchor Text에서 출현한 단어를 이용하여 사이트 내의 연관 문서를 찾는 방법을 제시하였다.

이 외에도 "The importance of prior probabilities for entry page search"[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, "하이퍼 텍스트의 가중치 조절과 링크 구조 분석 기법을 통한 검색 엔진 성능 개선"[4]에서는 In-링크와 Out-링크의 Anchor Text를 이용한 검색 방법을 제시하고 있다.

3. 제안하는 모델

본 논문은 선행 연구[15][16]의 모델에 사이트 가중치(Site Weight)를 추가한 모델을 제시하고, 실험 및 분석을 통한 결과를 보여준다. 사이트 가중치는 사이트의 대표 단어를 이용한 가중치이며, 사이트 가중치의 적용은 사이트의 대표 단어를 추출한 후, Query가 해당 문서가 포함된 사이트의 대표 단어일 때, 문서의 가중치를 높여준다.

3.1 링크들의 목록을 이용하여 Anchor Text 벡터 만들기

Anchor Text 벡터는 Anchor Text, Page Rank의 값 및 Link 정보를 결합하여 검색 성능을 높일 수 있도록 처리하여 만든 벡터이며, Anchor Text 벡터의 성능은 선행 연구인 "Anchor Text 정보와 링크 정보를 이용한 정보 검색 모델"[15]에서 보여주었다.

Anchor Text 벡터는 다음과 같은 방법으로 만들어진다.

- ▷ 한 문서를 가리키는 모든 Anchor Text를 합하여 하나의 Anchor Text 집합을 만든다.
- ▷ 벡터는 TF * IAF 방식을 이용하여 구한 값들의 집합이다.
- ▷ Anchor Text 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 Text 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. (논문에서는 10으로 제한 값을 두었다.)

▷ 한 Anchor Text에서 출현한 단어의 빈도는 출현했으면 1, 없으면 0으로 둔다. (논문에서는 여러 링크의 Anchor Text에서 동시 출현한 단어의 빈도가 가치가 있다고 간주하여 실험하였다.)

위에서 언급한 처리의 수식은 다음과 같다.

Step 1 : Page Rank의 값을 위한 정규화 수식

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

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
λ : 임의의 값 (실험에서는 4를 사용함)

Step 2 : Tf - Iaf 수식

$$Score(a_i, t_j) = tf_{ij} \times \log \frac{(N+1)}{af_j} \quad (2)$$

a_i = i번째 Anchor Text 집합
t_j = j번째 단어
tf_{ij} = i번째 Anchor Text 집합에서 j번째 단어가 출현한 빈도, tf_{ij} <= 255
N = 전체 Anchor Text 집합 수
af_j = j번째 단어가 출현한 Anchor Text 집합의 총 수

3.2 사이트의 대표어 구하기

사이트의 대표어를 구하기 위해 사이트의 외부 문서에서 오는 Anchor Text들을 각 사이트별로 추출하였으며, 실험을 통해 사이트 내부 문서간에 존재하는 Anchor Text는 가치가 적다는 것을 확인하였다. 추출된 단어 정보에서 출현한 모든 단어의 Frequency를 합쳐서 Total Frequency를 구한 후, **Total Frequency의 1% 이상의 빈도를 가진 단어들**만 대표어로 추출한다. 1%는 실험을 통해 구한 값이며, 사이트의 대표어를 구하는 처리 과정은 다음과 같다.

```
If(TermFreq / TotalFreq) * 100 >= 1 (1% 이상)
   해당 단어를 사이트 대표어로 추출
else
   해당 단어를 필터링
```

3.3 Query와 문서와의 유사도를 계산하는 방법

입력 Query와 문서와의 유사도는 다음의 과정을 통해 계산된다.

Step 1 : 사이트 대표어의 Frequency에 대한 처리

Query가 해당 문서가 포함된 사이트의 대표어라면 Site Value(사이트 가중치)의 값은 사이트의 외부 Anchor Text에서 나타난 대표어의 Frequency를 반영하여 구한다. Site Value(사이트 가중치)의 최대값은 10으로 설정하였다.

```
If( query가 해당 문서가 포함된 사이트의 대표어라면 )
   Site Value
   = log(사이트의 외부 Anchor Text에서 나타난 대표어의
   Frequency + 1)
else
   Site Value = 1
```

Step 2 : 문서 벡터를 계산하는 Tf-idf 수식

$$Score(d_i, t_j) = (tf_{ij} + w \times T_{i,j}) \times \log \frac{(N+1)}{df_j} \quad (3)$$

d_i : i번째 문서
t_j : j번째 단어
$tf_{i,j}$: i번째 문서에서 j번째 단어가 출현한 빈도 tf_{ij} <= 255
$T_{i,j}$: i번째 문서의 Title에서 j번째 단어의 출현 여부 (있으면 1, 없으면 0)
N : 문서의 총 수
df_j : j번째 단어가 출현한 문서의 총 수
w : 임의의 가중치

Step 3 : 해당 문서의 유사도를 계산하는 수식

$$Score(d_i) = s_i \times [\lambda \times f(Q, A_i) + (1 - \lambda) \times f(Q, D_i)] \quad (4)$$

d_i : i번째 문서
λ : 1보다 작은 임의의 값
s_i : i번째 문서의 Site Value (사이트 가중치)
Q : 질의어 벡터
A_i : i번째 문서의 Anchor Text 벡터
D_i : i번째 문서의 문서 벡터
f(x,y) : x와 y의 유사도를 계산하는 벡터 함수

3.4 사이트의 가중치가 하는 역할

선행 연구[15][16]에서 Anchor Text을 이용한 실험을 통해 Anchor Text의 장단점을 알게 되었다. Anchor Text 정보가 검색 성능에 향상을 가져오는 것은 실험을 통해 입증되었으나, 부작용도 생기는 것을 알게 되었다. 그 중 하나가 음란사이트, 신문, 포탈 사이트 등이 Query와 상관이 없음에도 상위에 검색되는 것을 볼 수 있었다. 특히, 음란사이트에 의한 부작용이 심하였다. 그 이유는 음란사이트들이 검색 결과 내에서 높은 순위로 검색되기 위해 "이효리","벅스뮤직" 등의 검색 빈도가 높은 단어를 Anchor Text에 사용하기 때문이다. 이런 Anchor Text의 부작용을 사이트 가중치가 완화시킨다. 이런 사이트의 경우 단어들의 빈도가 높더라도 실제적으로 전체 빈도에서 1% 이상의 빈도를 차지하는 단어가 되지 않기 때문에 Query와 사이트 대표어가 일치하는 사이트 내의 문서에 비해 낮은 가중치를 가진다.

사이트 가중치에 의한 부작용을 고려하여 사이트 가중치 자체의 값이 높더라도, 해당 문서에서의 벡터 계산에 의한 가중치가 높지 않다면 상위 문서로 검색되지 않도록 두 값을 곱하는 방식으로 처리한다. 실제 실험 시 벡터 연산에 의해 구해진 상위 100위 안의 문서들의 가중치는 100~1000 정도이며, 이는 상위 100위 정도 안에 존재하는 가치 있는 문서가 아니라면 사이트 가중치에 의해 10위 안에 들어갈 수 없다는 것이다.

4. 실험 및 결과

실험은 부산대학교 한국어정보처리연구실에서 수집한 1,000만 건 문서를 가지고 수행하였다.

본 논문에서는 사이트 가중치를 구하는 방법으로 사이트 대표어의 추출 및 가중치 부가를 사용한다. 사이트의 대표어는 사이트의 외부 문서에서 오는 Anchor Text들을 정보로 이용하며, 이 정보의 필터링을 통해 사이트의 대표어를 구하게 된다. 사이트의 대표어를 구하는 방법은 전체 단어 빈도에 대한 해당 단어의 상대빈도를 Threshold로 이용한다.

사이트의 대표어에 대한 가치를 평가하기 위하여 사이트의 대표어를 가지고 역파일을 구축한 후, 사이트의 대표어 정보만을 이용한 사이트 검색을 통해 성능 평가를 하였다. 검색 성능은 검색된 사이트들의 Query와의 관련 여부로 확인하였다.

[표 4.1]은 상대빈도의 비율을 조정하여 실험한 결과를 보여준다.

Threshold / 정확도	0.5 %	1 %	2 %	3 %	5 %	10 %	20 %
정확도(%)	71.8	85.8	86.6	87.5	87.7	88.0	90.5

[표 4.1] Threshold의 값에 따른 정확도

사이트 대표어 추출을 위한 Threshold의 값이 높아짐에 따라 정확도는 상승하나, 실제 검색되는 사이트의 수가 줄어든다. 즉, 추출되는 대표어의 개수가 줄어든다는 것이다. 실험을 통해 1~3%의 구간에서 추출되는 대표어의 개수와 성능이 적당하다고 판단하여, 1%를 선택하여 실험을 수행하였다.

[표 4.2]는 사이트 가중치를 부가한 모델과 다른 모델들의 성능을 보여준다.

모델 / 정확도	평균 정확도
DocScore	71%
DocScore + Title	76%
DocScore + Site	75%
PageRank73	78%
SiteRank73	77%
PageRank73$W_3$4	82%
SiteRank73$W_3$5	83%
PageRank73$W_3$4Site	**85%**
SiteRank73$W_3$5Site	**86%**

[표 4.2] 모델들의 성능표

DocScore (1) : 문서의 본문만을 이용한 검색 모델
DocScore + Title (2) : 문서의 본문과 Title Text를 이용한 검색 모델
DocScore + Site (3) : 문서의 본문과 사이트 가중치를 이용한 검색 모델
PageRank73 (4) : Anchor Text 벡터의 구축 시, Page Rank의 값을 이용하고 Anchor Text 벡터와 Document 벡터의 비중을 7:3으로 준 모델
SiteRank73 (5): Anchor Text 벡터의 구축 시, Site Rank의 값을 이용하고 Anchor Text 벡터와 Document 벡터의 비중을 7:3으로 준 모델
PageRank73$W_3$4 (6) : Anchor Text 벡터의 구축 시, Page Rank의 값을 이용하고 Anchor Text 벡터와 Document 벡터의 비중을 7:3으로 주고, Title의 가중치를 4로 준 모델
SiteRank73$W_3$5 (7) : Anchor Text 벡터의 구축 시, Site Rank의 값을 이용하고 Anchor Text 벡터와 Document 벡터의 비중을 7:3으로 주고, Title의 가중치를 5로 준 모델
PageRank73$W_3$4Site (8) : Anchor Text 벡터의 구축 시, Page Rank의 값을 이용하고 Anchor Text 벡터와 Document 벡터의 비중을 7:3으로 주고, Title의 가중치를 4로 주고, 사이트 가중치를 추가한 모델
SiteRank73$W_3$5Site (9) : Anchor Text 벡터의 구축 시, Site Rank의 값을 이용하고 Anchor Text 벡터와 Document 벡터의 비중을 7:3으로 주고, Title의 가중치를 5로 주고, 사이트 가중치를 추가한 모델

(1) (2), (3) (4) (5) 번 모델들의 정확도의 비교를 통해 사이트 가중치가 Anchor Text Vector나 Title Text보다는 검색 성능을 높이는 능력이 약한 것을 알 수 있으며, (6), (7), (8), (9) 번 모델들의 정확도를 비교를 통해, 사이트 가중치가 검색 성능을 높이는 능력이 Anchor Text Vector나 Title Text에 비해서 낮으나 전체 성능을 올리는 데에 기여를 한다는 것을 알 수 있다.
위의 모델간의 성능 비교를 통해 Anchor Text Vector는 약 6~7% 정도의 검색 성능을, Title Text는 약 4~6% 정도의 검색 성능을 향상시키는 것을 볼 수 있으며, 사이트 가중치는 약 3~4% 정도의 검색 성능을 향상시키는 것을 알 수 있다.

5. 결론 및 향후 과제

이 논문에서는 이전에 실험한 모델에 사이트의 가중치를 추가하여 검색의 성능을 개선하였으며, 여러 모델들과의 비교를 통해 각각의 정보들이 검색 성능에 어떠한 영향을 주는지를 보여주었다. 실험에서 나타났듯이 사이트의 가중치는 검색 성능의 향상을 가져오며, 특히 부적절한 목적을 위한 Anchor Text들에 의한 부작용을 줄일 수 있다.
향후 과제로는 수식에 사용된 가중치들에 대해 검색 성능을 가장 높일 수 있는 값을 추가 실험을 통해 구해야 하며, 사이트의 대표어를 추출하는 더 좋은 방법에 대해서도 연구를 해야 한다. 또한, 단어의 빈도만이 아니라 단어의 특징, 속성을 이용한 검색 모델에 대한 실험과 사용자의 요구에 따라 검색 모델을 다르게 하는 실험을 할 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.
[15] 한기덕, 정성원, 허희근, 이교운, 권혁철. Anchor Text 정보와 링크 정보를 이용한 정보 검색 모델. 한국정보과학회 봄 학술발표논문집, 2004.
[16] 이교운. 한국어 웹 문서의 구조적 정보와 색인어 정보를 결합한 검색 성능 향상. 부산대학교 박사 학위 논문, 2004.