사이트 검색을 위한 메타 검색

이여진○ 강인호 김길창
한국과학기술원 전자전산학과 전산학전공
{yjlee, ihkang, gckim}@csone.kaist.ac.kr

Metasearch for Website Finding

Yeojin Lee○ In-Ho Kang Gil Chang Kim
Division of Computer Science
Department of Electrical Engineering & Computer Science
KAIST

요 약

여러 검색 엔진이 낸 결과를 결합하여 성능의 향상을 얻고자 하는 정보 검색 방법을 메타 검색(metasearch)이라고 한다. 정보 검색에서의 사용자 요구가 다양화 되고 있지만, 기존의 메타 검색에 관한 연구는 이를 제대로 반영하지 못하고 웹 문서를 대상으로 검색(topic relevance task)한 결과를 결합하는 데에만 치중해 있다. 최근에는 사이트 검색(entry page finding task)만을 목적으로 한 시스템도 개발되고 있다. 본 논문에서는 사이트 검색 엔진들의 결과를 결합하는 메타 검색 방법을 제시한다. 웹 문서 검색 결과를 결합시에는 여러 검색 엔진에서 중복(overlap)하여 나타난 문서에 가중치를 두는 방법이 효과적이다. 하지만, 이 방법을 그대로 사이트 검색에 적용하면 웹 문서 검색에서와 같은 좋은 결과를 낼 수 없다. 본 논문에서는, 여러 검색 엔진에 중복하여 나타난 문서에 가중치를 두는 것 보다는 그 문서가 속한 사이트를 고려하여 사이트 단위로 중복된 정도를 반영하는 것이 사이트 검색 엔진의 결과를 결합하는 데 더 효과적임을 보인다.

1. 서론

많은 양의 정보가 산재한 웹 환경에서 사용자가 원하는 정보를 효과적으로 찾기 위한 다양한 검색 기법들이 제시되고 있다. 특정 상황에서 좋은 성능을 보이는 검색 기법이라 하더라도 다른 상황에서는 그렇지 못한 경우도 있다. 각 검색 기법의 장점을 살리고 단점을 보완할 수 있도록 여러 검색 기법을 결합하는 방법이 연구되는데, 이러한 검색 방법을 메타 검색(metasearch)이라 한다[1].

ProFusion[1]과 같은 메타 검색 엔진은 여러 검색 엔진(Altavista[2], Lycos[3], Yahoo[4])에 질의를 보내 그 결과로 나온 웹페이지들을 결합하여 사용자에게 순위화하여 보여준다. ProFusion은 각각의 독립적인 검색 엔진의 결과를 결합하기 때문에, 외부 메타 검색(external metasearch)이라 한다.

하나의 검색 엔진 안에서 다중 증거(*multiple evidence*)를 결합하여 결과를 낼 수도 있는데, 이러한 검색 방법을 내부 메타 검색(internal metasearch)이라 한다. 예를 들어, 문서의 내용, 인링크(in-link), 아웃링크(out-link), 키워드(keyword)와 같은 각각의 정보를 이용하면 다양한 방식의 인덱싱(indexing)과 검색 알고리즘이 가능하다. 이들 각각을 이용한 서브 검색 엔진(sub-search engine)의 결과를 결합하는 것이 바로 내부 메타 검색이다[2].

[1] http://www.profusion.com
[2] http://www.altavista.com
[3] http://www.lycos.com
[4] http://www.yahoo.com

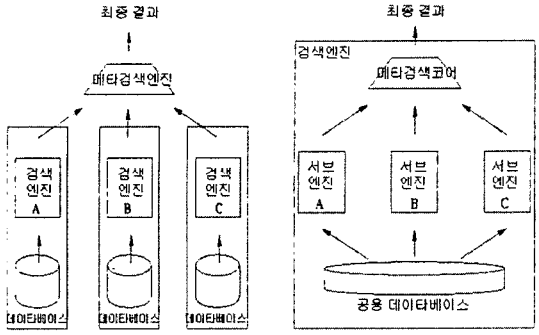

그림 1. 외부 메타 검색과 내부 메타 검색

Montague[2]는 그림 1과 같이 외부 메타 검색과 내부 메타 검색을 도식화하였다. Montague[2]에 따르면, 메타 검색 알고리즘은 외부 메타 검색에서보다 내부 메타 검색에서 더 효과적이다. 본 논문에서의 메타 검색 기법은 내부 메타 검색과 외부 메타 검색 모두에서 사용할 수 있는 것이나, 내부 메타 검색을 기반으로 하여 설명한다.

일반적으로 웹 문서들은 웹사이트를 단위로 하여 구성된다. 즉, 웹사이트는 하나의 주제를 담고 있는 여러 웹 문서들로 이루어지고, 각 웹 문서는 어떤 웹사이트의 일부분으로서 존재한다[5]. 일반적으로 웹사이트는 특정 주제와 관련있는 개인이나 단체가 운영한다. 사용자는 웹 사이트의 일부분인 웹 문서들을 원할 수도 있지만(topic relevance task), 웹 사이트 자체를 원하기도 한다(entry page finding task). 예를 들어, "디자인에서의 행동유도성"이라는 질의를 하는 사용자는 이 질의에 대해서 기술한 웹 문서를 가능한 한 많이 찾기를 원한다. 하지만, "한국과학기술원 산업디자인과 제품환경체계 연구실"이라는 질의를 하는 사용자는 해당 사이트의 입구(entry page)로 가는 것을 원한다.

이와 같이 검색 대상과 목적이 다양해지고 있지만, 기존의 메타 검색에 대한 연구는 웹 문서의 검색 결과를 어떻게 잘 결합할 것인가에 초점이 맞추어져 있다. 사용자가 사이트 검색을 원하는 경우, 이 결과를 결합하는 방법에 대해서는 제대로 연구되고 있지 않다. 최근에는 사이트 검색만을 위한 검색 엔진에 대해서도 충분히 연구되고 있기 때문에[2,4,5], 이들의 결과를 결합하여 더 좋은 성능을 얻는 것이 가능하다.

본 논문에서는 웹 문서 검색의 결과를 결합하는 것과 사이트 검색의 결과를 결합하는 것이 차이가 있음을 보인다. 웹 문서 검색의 결과를 결합하는 데에는 여러 검색 엔진에 중복해서 나타나는 문서에 가중치를 두는 방법이 이용되었다. 사이트 검색의 특성상 정답 문서의 개수는 웹 문서의 경우보다 작기 때문에, 이러한 방법을 사용하면 여러 검색 엔진에 중복해서 나타났지만 정답이 아닌 많은 문서들도 강조하게 될 가능성이 크다. 본 논문에서는 중복의 개념을 문서 단위가 아닌 사이트 단위로 확장할 필요가 있음을 보인다.

표 1. Fox & Shaw가 제안한 유사도 결합 함수

이름	결합 후의 유사도
combMIN	서로 다른 검색기가 산출한 유사도 중 최소값
combMAX	서로 다른 검색기가 산출한 유사도 중 최대값
combMED	서로 다른 검색기가 산출한 유사도의 중앙값
combSUM	서로 다른 검색기가 산출한 유사도의 총합
combANZ	combSUM ÷ (0이 아닌 유사도를 낸 검색기의 개수)
combMNZ	combSUM × (0이 아닌 유사도를 낸 검색기의 개수)

본 논문의 구성은 다음과 같다. 2장에서는 웹 문서 검색의 결과를 결합하는 데에 치중해 있던 현재까지의 메타 검색에 대한 관련 연구를 소개한다. 3장에서는 사이트 검색의 결과를 결합하는 메타 검색 모델을 제시한다. 4장에서는 실험을 통해 웹 문서 검색 결과를 결합하는 것과 사이트 검색 결과를 결합하는 것이 차이가 있음을 보이고, 검색 엔진의 사이트에 대한 선호도를 반영한 메타 검색 모델을 사이트 검색 결과를 결합하는 데 적용하여 그 효과를 보인다. 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

2.1. 결합 함수

Fox & Shaw[8]는 서로 다른 검색 방법을 쓰는 5개 검색 엔진의 결과를 결합하고자 하였다. 각 검색 엔진은 문서와 질의간의 유사도를 계산하여 유사도가 큰 순서대로 순위화하여 보여준다. 여러 검색 엔진의 결과를 바탕으로 메타 검색을 하기 위해서, 각 검색 엔진이 산출한 문서의 유사도를 결합해야 한다. 표 1의 combMIN, combMAX, combMED, combSUM, combANZ, combMNZ와 같은 함수가 문서의 유사도를 결합하는 데 쓰인다.

검색엔진 i에서 문서 d가 얻은 유사도를 $s_i(d)$라 하고, 문서 d를 결과로 낸 검색 엔진의 개수를 $n(d)$라 하면, combSUM, combANZ, combMNZ는 식 (1)과 같이 나타낼 수 있다.

$$score(d) = n(d)^r \sum_i si(d) \qquad \text{—— 식 (1)}$$

식 (1)에서 r=-1이면 combANZ, r=0이면 combSUM, r=1이면 combMNZ을 나타낸다[1].

Fox & Shaw[8]의 실험 결과, combSUM이 combMNZ에 비해 조금 더 좋은 성능을 보였고, 다른 결합 함수들은 메타 검색의 성능 향상에 별로 도움을 주지 못했다. 이들이 제안한 결합 함수는 이후 메타 검색의 기본 알고리즘이 되었다. 본 논문에서 제안하는 방식도 combSUM과 combMNZ를 기본으로 하여 비교할 것이다.

2.2. 유사도의 정규화

Lee[7]는 여러 검색 엔진이 매우 다른 범위의 유사도를 산출함을 발견하고, 결합 함수를 적용할 때 원래의 유사도를 그대로 사용하면 예상치 못한 결과를 낼 수도 있음을 지적했다. Lee[7]는 식 (2)를 사용하여 각 검색 엔진의 유사도 값을 [0,1] 범위내로 정규화한 후 결합 함수를 적용하였다.

$$\bar{s}(d) = \frac{s(d) - \min(s)}{\max(s) - \min(s)} \quad \text{—— 식 (2)}$$

Lee[7]의 실험 결과, Fox & Shaw[8]와는 달리 combMNZ가 combSUM보다 좋은 성능을 보였다.

Lee의 표준 정규화 방법 이외에도, 검색 엔진이 산출한 유사도의 총합을 1로 변환하여 [0,1]의 범위내로 유사도를 조정하는 합계 정규화 방법, 유사도 값의 평균을 0으로 변환하고 0을 중심으로 표준변차를 단위로 하여 유사도값을 재조정하는 평균중심 변환법 등이 있다[1]. 본 논문에서는 구현이 용이하면서도 효과적인 표준 정규화 방법을 사용한다.

2.3. 검색 엔진들의 결과 결합에 대한 근거

Lee[7]은 두 검색 시스템으로 메타 검색을 할 때 정답 문서와 비정답 문서가 두 검색 결과에 공통적으로 나타나는 비율을 계산하여, '서로 다른 검색 엔진은 서로 다른 집합의 정답이 아닌 문서를 검색하지만, 유사한 집합의 정답 문서를 검색한다'는 것을 발견하였다. combMNZ는 여러 검색 엔진이 결과로 제시한 문서에 가중치를 두기 위해서 combSUM 값을 조정한 것으로, 이러한 이론과 부합하는 결합 함수이다.

여러 검색 엔진에 공통으로 나온 문서가 높은 순위를 얻을 수 있도록 하는 combMNZ는 일반적인 웹 문서 검색의 결과를 결합하는 데는 가장 효과적이면서도 간단한 방법이다. 하지만 이것이 사이트 검색 엔진의 결과를 결합하는 데도 효과적인지에 대해서는 아직 연구되지 않았다.

3. 사이트 검색 엔진 결과의 결합

3.1. 중복 개념의 확장

웹 문서 검색의 결과를 결합할 때는 여러 검색 엔진에 공통적으로 나타나는 문서에 가중치를 두는 것이 효과적이었다. 하지만, 이러한 방법이 사이트 검색 엔진의 결과를 결합할 때도 좋은 성능을 보인다고 할 수는 없다.

TREC Web Track에서 사이트 검색에 해당하는 entry page finding task에 제출된 두 검색 시스템 tnout10epCAU와 IBMHOMER를 살펴보자. "미주리 대학교 세인트 루이스 캠퍼스의 국제 문제 연구소"로 방문하고자 하는 75번 질의 "Center for International Studies, University of Missouri-St. Louis"에 대한 정답은 문서 번호 WTX089-B46-211인 http://www.umsl.edu:80/services/cis/home.html이다. 그림 2는 이 질의에 대한 tnout10epCAU와 IBMHOMER의 결과를 1위부터 10위까지 부분적으로 나타낸 것이다. 정답 문서인 WTX089-B46-211는 IBMHOMER 검색 엔진에서는 1위로 검색되지만, tnout10epCAU에서는 검색되지 않았다. 이런 경우, 두 검색 엔진에 중복하여 나타난 문서에 가중치를 주는 combMNZ로 메타 검색을 하면, 정답 문서는 좋은 순위를 얻지 못한다. 오히려, 두 검색 엔진에 공통적으로 나타나기는 했지만 관련성이 적은 WTX084-B44-31, 즉 http://lawlib.slu.edu:80/과 같은 문서가 더 높은 순위를 얻게 된다.

이러한 문제를 해결하기 위해서는 사이트 검색에서의 중복의 단위를 문서가 아닌 사이트로 하는 것이 효과적이다. 두 검색 엔진에서 특정 문서 자체가 같은 것은 아니더라도 문서가 소속된 사이트가 같다면, 검색 엔진의 결과들 중에서 그 사이트에 속한 문서들에 가중치를 둘 수 있다. 한 사이트의 특성은 그 사이트에 속한 여러 문서에 퍼져 있을 것이므로, 그 퍼져있는 정보들을 모아서 고려할 필요가 있다. 그림 2에서 '*'로 표시된 시스템의 결과들은 http://www.umsl.edu/에 속한 문서들을 나타내고 있다. http://www.umsl.edu:80/services/cis/home.html이라는 문서는 한 검색 엔진에서만 나타난다. 하지만, 이 문서가 속해 있는 사이트인 http://www.umsl.edu/는 두 검색 엔진 모두에 자신의 소속 문서들을 내고 있다. 이러한 정보는 사이트 검색의 결과를 결합하는데 요긴하게 사용될 수 있다. 실제 사이트는 URL의 호스트(host) 이름으로 결정되는 것은 아니지만, 본 논문에서는 같은 호스트명을 가지는 문서는 같은 사이트에 속한다고 가정한다.

<tnout10epCAU>

순위 문서번호 유사도

1 WTX084-B44-316 2.705788
http://lawlib.slu.edu:80/

2 WTX029-B01-360 0.156744
http://www.slu.edu:80/departments/me

3 WTX040-B27-113 0.153045
http://www.smart-card.com:80/

4 WTX089-B46-68 0.148021
http://www.umsl.edu:80/services/ora/

5 WTX004-B24-219 0.142839
http://moe.med.yale.edu:80/index.html

6 WTX086-B40-213 0.132105
http://www.stjoan.midrealm.org:80/~cip

7 WTX090-B21-191 0.131021
http://www.umsl.edu:80/~intelstu/

8 WTX059-B17-60 0.103338
http://www.cus.wayne.edu:80/

9 WTX083-B49-255 0.089100
http://www.mdn.org:80/

10 WTX045-B38-198 0.085887
http://www.netwise.net:80/

<IBMHOMER>

순위 문서번호 유사도

1 WTX089-B46-211 842864.000000
http://www.umsl.edu:80/services/cis/home.html

2 WTX042-B27-174 841704.000000
http://www.umsl.edu:80/divisions/artscience/forlanglit/modGreek.html

3 WTX084-B44-316 829543.000000
http://lawlib.slu.edu:80/

4 WTX089-B46-62 827858.000000
http://www.umsl.edu:80/depts/depts.html

5 WTX090-B21-199 825389.000000
http://www.umsl.edu:80/services/cis/prof/prof.html

6 WTX090-B12-182 821596.000000
http://www.umsl.edu:80/services/academic/10scrtps.htm

7 WTX091-B17-325 819247.000000
http://www.umsl.edu:80/services/cis/prof/greek.html

8 WTX090-B21-180 818039.000000
http://www.umsl.edu:80/services/cis/research.html

9 WTX090-B31-131 817020.000000
http://www.umsl.edu:80/business/faculty/mis/janson.htm

10 WTX090-B21-195 815619.000000
http://www.umsl.edu:80/services/cis/truman.html

그림 2. "Center for International Studies, University of Missouri-St. Louis"라는 질의에 대한 두 검색 엔진의 1위부터 10위까지의 결과

표 2. 두 검색 엔진에 중복하여 나타난 것이 정답일 비율

	entry page finding (사이트 검색)	topic relevance task (웹 문서 검색)
$\frac{\lvert R \cap E_1 \cap E_2 \rvert}{\lvert E_1 \cap E_2 \rvert}$	0.103237	0.107774
$\frac{\lvert RS \cap ES_1 \cap ES_2 \rvert}{\lvert ES_1 \cap ES_2 \rvert}$	0.327931	0.142527

R : 정답 문서의 집합

E_n: 검색 엔진 n의 결과 문서의 집합

RS : 정답 문서가 속한 사이트를 원소로 갖는 집합

ES_n: 검색 엔진 n의 결과 문서가 속한 사이트를 원소로 갖는 집합

TREC Web Track의 topic relevance task와 entry page finding task에 제출된 검색 시스템 중 임의의 두 검색 시스템을 선정하여, 두 검색 엔진에서 공통적으로 나타나는 것이 정답일 비율을 조사하여 표 2에 나타내었다. 임의의 두 검색 시스템을 선정하는 작업은 100번 수행되었으며, 표 2에 나타난 값은 100번의 결과를 평균한 것이다.

표에서, 두 웹 문서 검색 엔진에 공통적으로 나타나는 문서 중 정답인 문서의 비율(0.103227)과, 두 사이트 검색 엔진에 중복하여 나타나는 문서 중 정답인 문서의 비율(0.107774)은 사이트 검색 엔진이나 웹 문서 검색 엔진이나 큰 차이가 없다. 그러나 사이트 단위로 그 비율을 조사하면 그 비율은 각각 0.327931, 0.142527로 차이가 크다. 이러한 특성은 문서 단위로 중복된 것에 가중치를 둘 것이 아니라, 사이트 단위로 중복된 것에 가중치를 두는 것이 더 효과적임을 보여준다. 꼭 문서 자체가 공통으로 출연한 것은 아니더라도, 그 소속 사이트가 두 검색 엔진에 공통으로 나온다면 그 사이트를 주의깊게 볼 필요가 있다.

3.2. 사이트 검색에서의 메타 검색 모델

Lee[7]는 웹 문서 검색의 결과를 결합할 때 여러 검색 엔진에 나타난 문서에 가중치를 주는 방식이 좋은 성능을 보인다고 하였다. combMNZ는 문서가 검색 엔진에서 얻는 유사도의 총합(combSUM)에 그 문서가 나타난 검색 엔진의 개수를 곱한 결합 함수로서, Lee의 이론에 충실한 결합 방법이다. 다음 식은 식 (1)과 동일하며, 값 위에 '¯'로 표시된 것은 식 (2)에서와 마찬가지로 [0,1] 범위로 정규화 되었음을 나타낸다.

표 3. TREC-2001 Web Track 데이터 분석

	질의의 갯수	질의당 정답 문서의 갯수	제출된 시스템의 갯수	평균 정확도		
				최소	최대	평균
topic relevance task (웹 문서 검색에 해당)	501-550 (50개)	67.26	97	0.02%	33.24%	15.48%
entry page finding task (사이트 검색에 해당)	1-145 (145개)	1.74	43	5.66%	74.04%	40.81%

$$combSUM(d) = \sum_i \overline{s}_i(d)$$

$$combMNZ(d) = n(d)\sum_i \overline{s}_i(d)$$

본 논문에서는 문서가 얻은 유사도의 총합 뿐만 아니라, 그 문서가 어느 사이트에 속하였는지에 대해서도 고려하고자 한다. 이를 위해서 combSUM에 사이트점수를 더하는 방법을 제안한다. 문서 d가 사이트 H에 속한다면, 메타 검색 엔진에서 문서 d가 얻게 되는 유사도는 다음과 같다.

$$score(d) = combSUM(d) + \overline{sitescore(H)}$$

사이트 H에 속한 문서들이 각 검색 엔진에 결과로 나왔을 때, 그 문서들이 각 검색 엔진에서 얻은 유사도의 총합을 사이트 H의 사이트 점수라고 한다.

$$sitescore(H) = \sum_i \sum_{\{d|\forall d \in H\}} \overline{s}_i(d)$$

즉, 같은 사이트에 속하는 문서는 같은 값의 사이트 점수를 가진다. 유사도를 정규화하는 식 (2)에서와 마찬가지로 사이트 점수도 [0,1] 범위 내로 정규화하여 사용한다.

$$\overline{sitescore(d)} = \frac{sitescore(d) - \min(sitescore)}{\max(sitescore) - \min(sitescore)}$$

4. 실험

4.1. 실험 데이터

TREC(Text REtrieval Conference)[5]은 NIST(US National Institute of Standards and Technology)에 의해 매년 개최되는 문서 검색에 관한 학술 대회이다. 그 중 Web Track은 웹 상에서의 정보 검색 기

[5] http://trec.nist.gov/

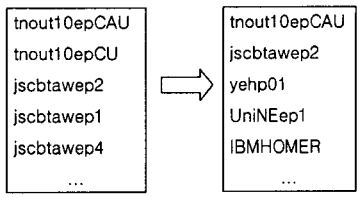

그림 3. 비슷한 검색 기법을 사용한 시스템 제외

량을 겨루는 대회로서, 약 1,690,000개의 웹 페이지로 구성된 10G의 코퍼스를 대상으로 하여 웹 정보 검색을 하도록 한다. TREC Web Track에서는 질의와 그에 해당하는 정답을 제공한다. 이를 바탕으로 하여 대학, 산업체 등에서는 검색 시스템을 만들어 제출한다. 이전까지의 TREC Web Track에는 웹 문서 검색에 해당하는 adhoc task만 있었으나, 2001년부터는 사용자 요구의 다양성을 고려하여 topic relevance task(이전의 adhoc task에 해당)이외에 사이트 검색에 해당하는 entry page finding task를 추가 하였다[3]. 각 검색 시스템은 한 질의에 대해서 topic relevance task의 경우에는 최대 1000개, entry page finding task의 경우에는 최대 100개까지 결과를 내고 있다. 본 논문에서는 TREC-2001 Web Track의 topic relevance task와 entry page finding task에 제출된 시스템들의 결과를 대상으로 메타 검색을 수행한다.

표 3은 본 논문의 실험에 사용된 데이터를 설명한다. 웹 문서 검색의 경우 질의당 정답 문서의 개수가 67.26개, 사이트 검색의 경우에는 1.74개이다. 사이트 검색의 경우 질의에 적합한 문서의 개수는 대개는 한 개이고, 그렇지 않은 경우에는 대부분 한 사이트의 미러 사이트이거나 URL 분기(redirection)에 의한 중복 문서로 정답 개수가 두 개 이상이 된 것이다.

4.2. 실험 방법

TREC-2001 Web Track의 시스템들을 이용할 때, 동일한 기관에서 제출한 검색 엔진은 비슷한 검색 기법을 사용한다고 가정하고 그 기관에서 제출한 시스템 중 가장 좋은 성능을 보이는 검색 엔진만을 남기고 나머지는 제외하였다. 그림 3은 entry page finding task에 제출된 시스템 중 성능이 우수한 검색 엔진 5개를 나타낸 것인데, 이것들은 모두 2개의 기관에서 제출한 것이다. 이와 같이 비슷한 검색 기법을 사용한 것으로 보이는 같은 기관의 시스템 중에서 최고 성능을 보이는 것만을 남기고 실험하였다. topic relevance task에는 유사 검색 엔진을 제거한 30개 시스템만을 이용하였고, entry page finding

task에는 16개의 시스템만을 이용하였다. 실험은 다음과 같은 두 가지 방법을 사용하여 실행하였다.

4.2.1. 검색 엔진 무작위 선택 실험

검색 엔진 무작위 선택 실험은 입력이 되는 검색 엔진의 특성에 영향을 받지 않고 실험 결과의 신뢰성을 높이기 위해 수행하였다. 검색 시스템 중 n개(단, n∈{2,3,4,5})를 임의로 고른 후에 그 결과를 결합하는 과정을 100번 시행하여 그 결과를 평균하였다.

<n개의 검색 엔진 결과를 결합하는 무작위 선택 실험>
1. 임의의 검색 시스템 n개를 얻는다
2. 결합 함수를 사용하여 각 시스템의 결과를 합친다.
3. 합친 메타 검색 결과의 성능을 평가한다.
4. 1~3의 과정을 100번 반복한다.
5. 3의 메타 검색 결과 100개를 평균하여, 결합 함수의 일반적인 성능을 얻는다.

표 3에 나타난 것처럼, 각 검색 시스템의 성능은 웹 검색의 경우 0.02%~33.24%, 사이트 검색의 경우 5.66%~74.04%로 다양하게 분포하고 있다. 검색 엔진 무작위 선택 실험을 통해, 다양한 성능의 검색 엔진을 결합한 결과를 알아볼 수 있다.

4.2.2. 최고 성능 검색 엔진 선택 실험

최고 성능 검색 엔진 선택은 검색 시스템을 성능별로 정렬한 후, 최고의 성능을 보이는 n개(단, n∈{2,3,4,5})의 시스템을 결합하는 실험이다. 예를 들어, 그림 3에 나타난 검색 엔진 중 최고 성능을 보이는 tnout10epCAU부터 시작해서 차례대로 n개 선정해 그 결과를 결합하게 된다.

4.3. 성능 평가

본 논문에서는 검색 엔진의 성능을 측정하기 위해 평균 정확도(average precision)를 사용한다. 평균 정확도란, 검색 엔진의 결과로 나온 리스트에서 정답 문서가 나타날 때 마다 그 상위 순위까지의 정확도를 측정하여 평균한 것이다. 정답 문서의 집합을 R이라 하고 문서 d가 검색 엔진에서 얻은 순위를 $r(d)$라고 하였을 때, 평균 정확도는 다음과 같이 나타낼 수 있다[1].

$$P_{avg} = \frac{1}{|R|} \sum_{d \in R} \frac{|R \le r(d)|}{r(d)}$$

평균 정확도로 검색 엔진의 성능을 측정하면, 정답 문서를 높은 순위에 보여주는 검색 엔진이 더 높은 평균 정확도 값을 가지게 된다[6].

메타 검색을 통해 얼마만큼의 성능 향상이 있었는지를 알 수 있도록, 입력 검색 시스템 중에서 가장 좋은 성능을 보이는 것에 비례해 얼마나 성능이 개선되었나를 측정하였다.

n개의 입력 검색 시스템 중 가장 좋은 성능을 보이는 시스템의 평균 정확도를 P_b라 하고, 그 n개의 시스템을 결합한 메타 검색 엔진의 평균 정확도를 P_f라 하였을 때, 향상치 I를 다음과 같이 나타낸다.

$$I = \frac{P_f - P_b}{P_b}$$

향상치가 0이면 메타 검색 엔진의 성능이 입력 시스템 중 가장 높은 평균 정확도를 가지는 검색 엔진의 성능과 같음을 의미하고, 음수이면 성능 향상이 없었음을 나타내며, 양수이면 메타 검색으로 성능이 향상되었음을 나타낸다.

4.4. 실험 결과

4.4.1. 웹 문서 검색과 사이트 검색의 메타 검색

웹 검색과 사이트 검색의 결과를 combSUM과 combMNZ 함수를 사용하여 결합하였다. 검색 엔진 무작위 선택 실험과 최고 성능 검색 엔진 선택 실험을 시행한 후 그 향상치를 측정하여 그림 4에 나타내었다.

문서 단위로 중복의 정도를 살펴 보아 여러 검색 엔진이 결과로 낸 문서에 가중치를 두는 combMNZ는 웹 문서 검색의 결과를 결합하는 데는 효과적이었다. 하지만, 사이트 검색의 결과를 결합할 때 combMNZ를 적용하면, 웹 검색에서와 같은 효과를 얻지 못하고 오히려 가중치를 주지 않은 combSUM보다도 낮은 성능을 보인다. 또한, 사이트 검색 결과를 결합한 것은 웹 검색 결과를 결합한 것에 비해서 향상치가 그리 크지 않으며, 검색 엔진 무작위 선택 실험에서는 심지어는 향상치가 0이하인 경우도 많다. 향상치가 0이하라 함은, 기본 검색 엔진 중 최고 성능을 보이는 것보다도 메타 검색 엔진의 결과가 더 낮음을 나타내는 것으로 이와 같은 메타 검색 방법이 효과적이지 못하다는 사실을 나타낸다. 웹 문서 검색의 결과를 결합하기 위해 사용되었던 기존의 메타 검색 방법을 그대로 사이트 검색에 이용하면 좋은 성능을 내지 않는다. 사이트 검색 엔진의 결과를 결합하는 데에는 기존의 것과는 다른 함수가 필요함을 알 수 있다.

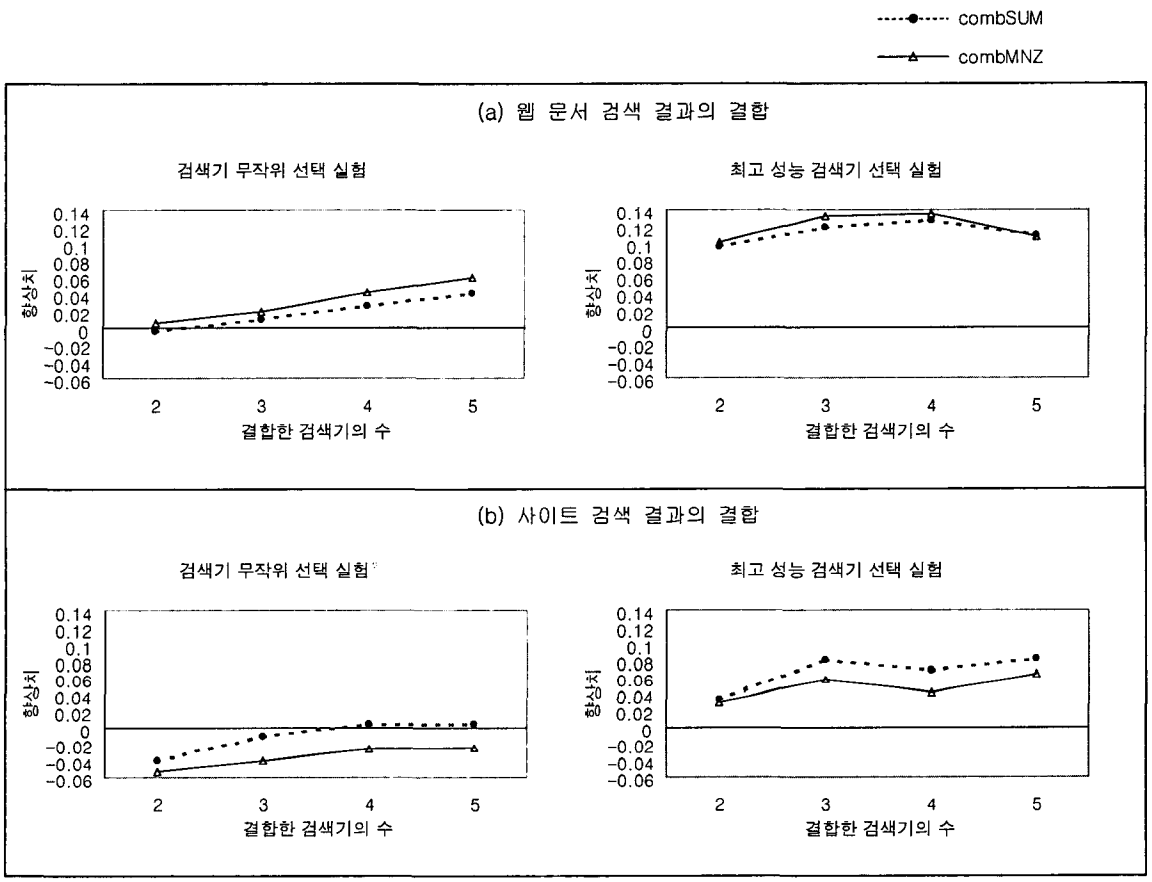

그림 4. 웹 문서 메타 검색과 사이트 메타 검색에 combSUM, combMNZ를 이용한 결과

4.4.2. 사이트 검색 결과의 결합

본 논문에서 제안한 함수로 사이트 검색의 결과를 결합하였다. 앞서와 마찬가지로 검색 엔진 무작위 선택 실험과 최고 성능 검색 엔진 선택 실험을 하였다. 그림 5는 combMNZ, combSUM과 함께 본 논문에서 제안한 함수를 사용한 결과를 나타낸다.

사이트 점수를 고려한 이 방법은 combMNZ와 combSUM에 비해 월등히 좋은 성능을 보인다. 이로써, 사이트 검색 결과를 결합할 때에는 문서 단위로 중복 정도를 고려하는 것보다는 사이트 단위로 중복 정도를 고려하는 것이 더 효과적임을 알 수 있다.

5. 결론 및 향후 연구

기존의 메타 검색 기법은 웹 문서 검색 엔진의 결과를 결합하는 데 치중해 있다. 웹 문서 검색 엔진 결과를 결합할 때는 여러 검색 엔진에 출현한 문서에 가중치를 두는 combMNZ가 뛰어난 성능을 보인다고 알려져 있다.

본 논문에서는 사이트 검색 엔진의 결과를 결합하는 것이 웹 문서 검색 엔진의 결과를 결합하는 것과 다르다는 것을 보였다. 사이트 검색은 웹 문서 검색보다 정답의 개수가 훨씬 작기 때문에, 여러 검색 엔진에 나타난 문서에 가중치를 주는 웹 문서 메타 검색의 휴리스틱을 사이트 검색에 사용하면, 정답이 아닌 문서의 순위가 오르게 된다. 따라서 여러 검색 엔진에 공통으로 나타난 문서에 가중치를 주는 combMNZ는 사이트 검색 엔진의 결과를 결합하는 데에는 부적합하다. 오히려 가중치를 주지 않은 combSUM보다도 나쁜 성능을 보인다.

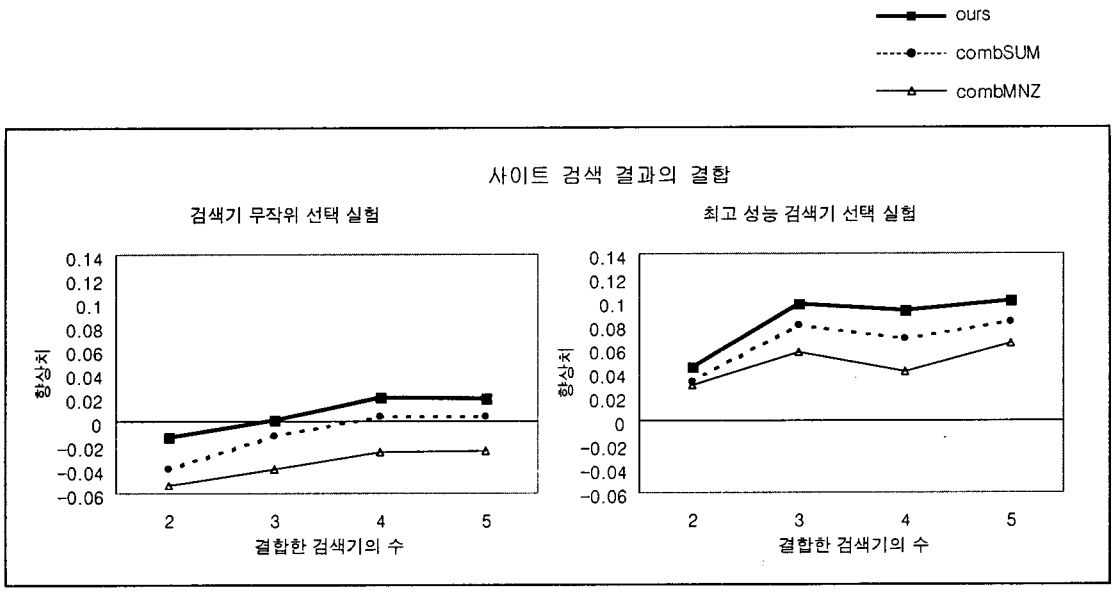

그림 5. 본 논문에서 제안한 함수로 사이트 검색 결과를 결합

사이트 검색의 결과를 결합할 때는, 문서가 어느 사이트에 속하는가를 고려하여, 여러 검색 엔진에 문서를 많이 낸 사이트에 가중치를 두는 것이 효과적이다. 사이트 단위로 중복의 개념을 확장한 함수를 이용하여 사이트 검색의 결과를 결합하면 기존에 제안된 combMNZ, combSUM 등의 함수보다 효과적으로 메타 검색을 할 수 있다.

본 논문에서 제안한 결합 함수는 10개 이하의 비교적 적은 개수의 검색 엔진 결과를 결합한다고 가정하였기 때문에, 각 문서의 combSUM 결과를 다시 정규화하지는 않았다. 그래서 많은 검색 엔진의 결과를 결합하면 상대적으로 사이트점수가 반영되는 비율이 작아질 수 있다. 이러한 문제도 해결해야 할 과제이다. 본 논문에서 제안한 함수 이외에도, 중복의 개념을 문서가 아닌 사이트 단위로 확장을 한 여러 형태의 메타 검색 함수가 가능할 것이다. 앞으로는 효과적인 결합 함수를 고안하는 연구도 해야 할 것이다.

6. 참고 문헌

[1] M. Montague, "Metasearch: Data fusion for document retrieval", PhD thesis, Dartmouth College, 2002.

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

[3] D. Hawking and N. Craswell, "Overview of the TREC-2001 Web Track", Proceedings of the 10th Text REtrieval Conference (TREC-2001), 2001.

[4] T. Westerveld, W. Kraaij and D. Hiemstra, "Retrieving Web Pages using Content, Links, URLs and Anchors", Proceedings of the 10th Text REtrieval Conference (TREC-2001), 2001.

[5] 장중식 외, "웹 환경에서의 홈페이지 검색 시스템", 제13회 한글 및 한국어 정보처리 학술대회, pp.70-75, 2001.

[6] R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval", Addison-Wesley, 1999.

[7] J.H. Lee, "Analyses of Multiple Evidence Combination", Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.267-275, 1997.

[8] E.A. Fox and J.A. Shaw, "Combination of multiple searches", Proceedings of the 2nd Text REtrieval Conference (TREC-2), pp.243-252, 1994.