메타서치엔진에서 네트워크의 트래픽을 줄이기 위한 검색엔진의 선택 및 검색문서의 수 결정

(Selection of Search Engine and the number of documents in Meta Search Engine to reduce network traffic)

이 진 호*, 박 선 진**, 박 상 호**, 남 인 길***
(Jin-Ho Lee · Sun-Jin Park · Sang-Ho Park · In-Gil Nam)

요 약 메타서치엔진에서 탐색하게 될 검색엔진의 수를 줄이거나, 각 검색엔진에서 반환할 문서의 수를 줄임으로서 메타서치엔진이 사용자에게 제공하는 전체 문서의 수를 줄여 네트워크상의 불필요한 트래픽을 감소시키면서 검색결과는 사용자의 질의어에 대한 적합도를 유지하는 방안을 제안하였다. 현재 많이 사용되는 검색엔진과 가장 검색 빈도수가 높은 검색어를 이용하여 검색엔진을 선택하는 방법과 검색문서의 수를 결정하는 방법을 실험하였다.

Abstract The decision method for the selection of search engine and the number of returned documents for meta search engine proposed in this paper could provide a solution to reduce network traffic and to maintain the precision ratio. The experiments are performed to evaluate the proposed scheme using currently popular search engines and most frequently used queries.

1. 서 론

정보 제공자의 수가 증가함에 따라 방대한 자료들이 인터넷에 연결되어지고 있고, 여기에 검색 가능한 정보량이 급속도로 증가하고 있어 사용자들은 효율적인 정보검색을 위하여 검색엔진을 이용한다. 정보검색을 위한 검색엔진은 사용자로부터 부여받은 질의어를 분석하여 질의어와 관련된 문서를 제공한다. 이러한 검색엔진의 요건으로는 첫째 많은 정보를 가지고 있어야 한다. 인터넷 웹사이트상의 많은 정보란 검색 가능 영역 안에 보유하고 있는 데이터베이스의 양이다. 즉 데이터베이스가 가지고 있는 웹페이지의 보유개수를 말한다. 둘째 검색엔진이 보유하고 있는 데이터베이스의 내용 및 색인의 갱신주기가 빨라야 한다. 정보가 빠른 주기로 변하듯 검색엔진의 결과가 사용자에게 신뢰를 주기 위하여 검색엔진의 갱신주기도 빨라야 한다는 것이다. 셋째 데이터베이스 내용의 신뢰성이 높아야 한다. 일반 사용자들이 검색엔진의 출력 결과 중 열람해 보는 페이지 수는 10~30개 안팎인 것으로 알려져 있으므로 [1,2], 검색 결과 앞부분에 출력된 결과물이 얼마나 믿을 만한가 하는 것이다. 그 외 검색 속도가 빨라야 한다는 것이다. 그러므로 검색엔진의 성능을 좌우하는 가장 큰 요소는 바로 보유하고 있는 데이터베이스의 양과 검색작업 내용의 신뢰성 여부에 있다고 하겠다.

그러나 하나의 검색엔진을 사용하여 필요한 정보를 얻기가 어려운 경우가 많이 있어 최근에는 자체적으로 검색로봇이나 데이터베이스를 가지고 있지는 않지만, 다수의 검색엔진을 한번에 검색하여 하나의 통일된 형태로 자료를 제공하는 메타검색기 또는 통합검색기가 등장하게 되었다 [1,3]. 메타검색엔진은 데이터베이스를 가지고 있지 않고 다수의 검색엔진으로부터 정보를 가져오게 되므로 정보의 양은 증가하지만 중복된 자료 및 부적절한 자료가 많아지게 되어 네트워크의 트래픽이 필요 이상으로 증가하게 된다. 그래서 각 검색엔진의 자원을 낭비하며 사용자가 불필요한 문서들을 열람하게 되는 확률이 높아진다.

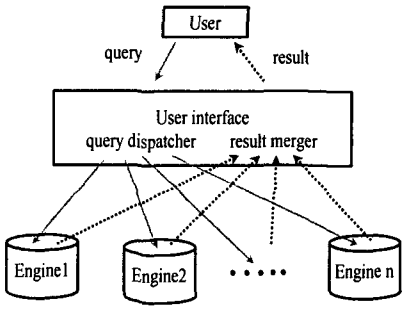
*경일대학교 공과대학 컴퓨터공학과 교수
**안동대학교 정보통신공학과 조교수
***대구대학교 컴퓨터정보공학부 교수

본 논문은 메타서치엔진에서 탐색하게 될 검색엔진의 수를 줄이거나, 각 검색엔진에서 반환할 문서의 수를 줄임으로서 메타서치엔진이 사용자에게 제공하는 전체 문서의 수를 줄여 네트워크상의 불필요한 트래픽을 감소시키고, 검색결과는 사용자의 질의어에 대한 적합도를 유지하는 방안을 제안한다. 메타서치엔진은 크게 3단계로 나누어 정보검색의 효율을 높일 수 있는데 첫 번째는 어떤 검색기를 선택하느냐 하는 문제이다. 두 번째는 선택된 검색기로부터 어떤 문서를 가지고 올 것인가 하는 문제이고, 세 번째는 가지고온 문서를 어떤 순서로 정렬하여 보여주는가 하는 문제이다 [4]. 본 연구에서는 메타서치엔진에서의 정보검색 효율을 높일 수 있는 첫 번째 단계로 사용자의 질의어에 대한 검색엔진의 적합성을 계산하여 검색엔진을 선택하거나 검색엔진으로부터 가지고 올 문서의 수를 줄임으로써 네트워크 상의 필요 이상의 트래픽을 줄이고 검색효율을 높일 수 있는 방안을 제안한다.

2. 메타서치엔진을 위한 검색엔진의 선택 및 검색문서의 수 결정

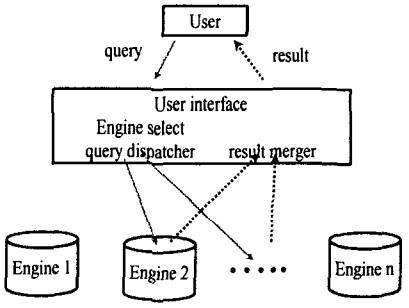
2.1 메타서치엔진

메타서치엔진의 일반적인 구조는 <그림 1>에서와 같이 사용자가 질의어를 주게되면 사용자 인터페이스에서 질의어를 각 검색엔진으로 넘겨주어 각 검색엔진에서 가지고 있는 질의어와 부합되는 문서를 탐색하게 된다. 탐색이 끝나면 각 검색엔진은 그 결과를 메타서치엔진에 돌려주게 된다 [4,5]. 메타서치엔진에서는 각 검색엔진으로부터 받아온 결과들을 병합하여 사용자에게 돌려주게 되는데 이러한 결과들을 병합할 때 각 검색엔진으로부터 가져온 중복된 자료를 제거하고 순위를 재 부여하게 되는 것이 일반적인 메타서치엔진의 구조가 된다. 사용자는 이러한 메타서치엔진을 사용함으로써 검색시간을 줄이고 이곳 저곳 검색엔진을 옮기면서 질의를 할 필요 없이 하나의 조작으로서 다수의 검색엔진에서 보유한 데이터베이스를 검색할 수 있어 다량의 정보를 찾을 수 있고 여러 검색엔진의 결과를 한눈에 볼 수 있으나 다소 검색시간이 많이 걸리고 네트워크의 트래픽이 증가하는 단점이 있다.



<그림 1>. 일반적인 메타서치엔진의 구조

메타서치엔진은 다수의 검색엔진으로부터 정보를 가져와서 통합하게 됨으로써 정보의 양이 많아지게 된다. 이러한 정보의 증가는 중복된 자료나 부적절한 자료가 많아지게 되며 네트워크 상의 트래픽을 증가시킨다. 네트워크의 트래픽을 감소시키고 중복된 자료나 부적절한 자료를 최소화하기 위해서는 탐색할 검색기의 선택과 각 검색기의 적합성에 따라 반환할 문서의 수를 결정하여 메타서치엔진에 반환되는 문서의 수를 감소시키고 질의어에 대한 각 문서의 적합도를 높여야 한다. <그림 2>는 사용자의 질의를 받아 검색엔진을 선택하고 선택된 검색엔진에게만 질의를 보내고 그 결과를 통합하여 사용자에게 검색 결과를 넘겨주는 효율적인 메타서치엔진의 구조를 보여주고 있다 [4].



<그림 2>. 효율적인 메타검색엔진의 구조

2.2 질의어에 대한 검색엔진의 평가

메타서치엔진은 자체 데이터베이스가 없으며 질의에 대한 문서검색을 위하여 다수의 검색엔진으로부터 정보를 가져오게 되므로 정보의 양이 많아지게 된다. 그러나 메타

서치엔진이 검색한 문서 중 사용자의 질의에 적합하지 못한 문서가 많이 포함되어 있는 경우가 빈번하게 나타난다. 그러므로 질의에 대한 최적의 결과 값을 돌려주기 위해서 사용자가 확인하는 문서의 수에 대하여 적합한 문서의 비가 커야 한다. 적합한 문서의 비를 높이기 위해 검색한 질의에 가장 적합한 검색엔진을 선택하는 것이 중요하다. 검색엔진의 선택은 질의에 대한 반환되는 문서의 근접도에 의하여 평가되고 검색엔진의 유용도(usefulness)와 반환시간에 따른 적합도에 의하여 이루어진다 [4-9].

메타서치엔진에 사용자가 검색하고자하는 질의어 q_j에 대한 검색엔진 S_i의 유용도(usefulness)는

$$usefulness(q_j, S_i) = \sum_{j=1}^{k} CVV_j \times df_{i,j} \quad (1)$$

여기서 $df_{i,j}$는 검색엔진 S_i에 검색한 질의어에 대한 전체 반환문서의 수이고, k는 검색한 질의어에 대한 텀의 수이다.

질의어 q_j에 대한 CVV(Cue Validity variance) CVV_j는

$$CVV_j = \frac{\sum_{i=1}^{N}(CV_{i,j} - ACV_j)^2}{N} \quad (2)$$

여기서 N은 검색엔진의 수이고 ACV_j는 질의어 q_j에 대한 평균 CV(cue validity)이며 질의어 q_j에 대한 $CV_{i,j}$는

$$CV_{i,j} = \frac{df_{i,j}/n_i}{\frac{df_{i,j}}{n_i} + \frac{\sum_{k \neq i}^{N} df_{k,j}}{\sum_{k \neq i}^{N} n_k}} \quad (3)$$

여기서 n_i는 검색엔진 S_i의 총 문서의 수이고 $df_{i,j}$는 검색한 질의에 대한 총 결과 수이며 N은 검색엔진의 수이다. n_k는 검색엔진 S_i을 제외한 검색기 S_k의 총 문서의 수가되며 $df_{k,j}$는 검색엔진 S_i을 제외한 다른 검색엔진 S_k가 가지고 있는 총 문서의 수가된다.

검색엔진에서 문서를 반환하는 반환시간(Presentation Time)은 각 검색엔진에 질의를 하나 보내고 그 질의에 대한 30개의 결과를 반환하는데 걸리는 평균 응답 시간으로 측정하였다.

$$Presentation\ Time(t, S_i) = C \times \frac{1}{\sqrt{\frac{Rt_{i,j}}{\sum_{i=1}^{N} Rt_{i,j}}}} \quad (4)$$

여기서 C는 반환시간을 조절하기 위한 상수이며 N은 검색기의 수이고, $Rt_{i,j}$는 하나의 질의에 대한 30개의 결과의 수를 반환하는 평균 시간이다. 사용자의 질의결과를 중시하고 네트워크의 부하 량은 중요시 여기지 않는다면 상수 C를 0으로 설정하면 된다.

사용자가 검색한 질의에 대한 검색엔진의 적합도(Fitness)는 질의에 대한 검색엔진의 유용도와 네트워크의 처리 속도를 나타내는 반환시간의 합이다.

$$Fitness(q_j, S_i) = usefulness(q_j, S_i) + Presentation\ Time \quad (5)$$

여기서 $usefulness(q_j, S_i)$는 사용자가 질의한 질의에 대한 검색엔진 S_i의 유용도이며, $Presentation\ Time(t, S_i)$는 검색엔진 S_i에 질의를 하고 질의한 결과를 30개까지 반환되는 반환시간이다.

2. 3 검색엔진의 선택 및 반환문서의 수 조정

적합도에 따라 검색엔진을 선택하고 각 검색엔진에서 가지고 올 반환문서의 수를 산정 하는데 네트워크의 트래픽을 줄이기 위하여 검색엔진을 배제할 경우는 적합도가 제일 적은 검색엔진을 배제하고, 각 검색엔진에서 가지고 올 반환문서의 수는 아래 식을 적용하여 구한다.

$$Return\ No(q_j, S_i) = \frac{Fitness_i}{\sum_{j=1}^{N} Fitness_j} \times m \quad (6)$$

여기서 m은 각 검색기에서 가져와 사용자에게 돌려줄 총 반환문서의 수가되며 $Fitness_i$는 사용자가 검색한 질

의에 대한 검색엔진 S_i의 적합도를 나타내며 $Fitness_j$는 각 검색엔진에서 얻은 적합도의 합이 된다. 여기서 N은 선택된 검색엔진의 수가된다.

사용자 질의어 q_j에 대한 검색엔진 S_i의 적합도를 가지고 다음과 같은 다섯 가지방법으로 검색엔진을 평가하였다.

● 방법1(단순적용) : 각 검색엔진에서 가져올 반환문서의 수를 동일하게 정하여 모든 검색엔진으로부터 문서를 가져온다.

● 방법2(결과조정) : 얻어진 적합도를 식(6)에 적용 각 검색엔진에서 반환하게 될 문서의 수를 달리하여 모든 검색엔진으로부터 문서를 가져온다.

● 방법3(엔진배제) : 적합도가 제일 낮은 검색엔진을 하나를 배제하고 검색엔진에서 가져올 반환문서의 수를 동일하게 정하여 문서를 가져온다.

● 방법4(배제&조정) : 적합도가 제일 낮은 검색엔진 하나를 배제하고 식(6)을 이용하여 반환하게 될 문서의 수를 달리하여 검색엔진으로부터 문서를 가져온다.

● 방법5(m조정) : 검색엔진을 배제하지 않고 식(6)을 이용하여 반환하게 될 문서의 수를 달리하여 검색엔진으로부터 문서를 가져온다.

2. 4 검색결과의 평가

사용자 질의어 q_j에 대한 메타서치엔진의 검색결과를 평가하기 위해서 일반적으로 사용자 질의에 대한 적합율(*Precision Ratio*)을 구하는 식(7) [2,10]을 이용하여 검색문서를 평가한다.

$$Precision\ Ratio = \frac{\sum_{i=1}^{N} D_i}{m} \times 100 \qquad (7)$$

여기서 m은 총 반환문서의 수이고 D_i 는 적합한 문서의 수이다. 그리고 N은 선택된 검색엔진의 수가된다.

3. 실험 및 고찰

본 논문에서는 제안된 검색엔진 선택방법을 실험하기 위하여 첫 번째 질의어마다 각 검색엔진의 유용도를 구하여 메타서치엔진에서 검색할 검색엔진의 선택과 반환할 문서의 수를 결정하였고, 두 번째 검색엔진에서 빈도수가 높은 질의 100개의 인덱스를 만들어 유용도을 구하여 그 평균치를 메타서치엔진의 갱신주기 까지 적용하는 방법이다. 실험을 위하여 사용된 검색엔진은 질의에 대한 검색엔진에서 가지고 있는결과의 수를 알 수 있는 AltaVista(AV) Excite(EX)와 Infoseek(IS), Northern Light(NL)를 사용하였다. 사용자가 검색한 질의에 대하여 검색엔진을 선택하거나 검색엔진에서 가지고 올 문서의 수를 결정하여 적용전과 적용한 후의 결과를 평가하기 위하여 Search Engine Watch에서 제공하는 빈도수가 많은 질의 100개중 가장 검색빈도수가 많은 상위 5개의 검색어로 각 검색엔진에서 질의와 관련된 문서의 인덱스를 <표 1>과 같이 구하였다.

<표 1>. 질의에 대한 인덱스

(단위: 만개)

query		NL	AV	EX	IS	합계
1	game	977	1,506	158	267	2,908
2	travel	740	2,458	169	253	3,620
3	music	847	3,300	239	282	4,668
4	sport	1,167	1,322	52	233	2,774
5	yahoo	235	809	64	36	1,144
전체문서의수		17,000	15,000	12,500	7,500	52,000

<표 1>의 데이터와 식 (1) - (3)을 이용하여 질의에 대한 검색엔진의 유용도를 <표 2>와 같이 구하였다. 네트워크의 속도를 나타내는 반환시간은 각 검색엔진에 질의 하나를 보내고 그 질의에 대한 문서를 30개까지 반환하는데 걸리는 평균 응답시간을 측정하여 식(4)에 적용하고 이때 상수 C는 1로 설정하여 <표 3>를 구하였다. 질의에 대한 검색엔진의 유용도와 네트워크 의 처리 속도를 나타내는 반환시간을 식(5)에 적용하여 <표 4>과 같이 사용자가 검색한 질의에 대한 검색엔진의 적합도를 구하였다.

적합도의 결과를 식(6)에 적용하여 검색엔진을 선택하거나 각 검색엔진에서 가지고 올 반환문서의 수를 다음과 같은 다섯 가지의 방법으로 각 질의에 대하여 검색엔진을 선택하고 반환문서의 수를 조정하였다.

● 방법1(단순적용) : 각 검색엔진에서 가져올 반환문서의 수를 30개씩으로 정하여 모든 검색엔진으로부터 120개의 결과를 가져온다.

● 방법2(결과조정) : 얻어진 적합도를 식(6)에 적용 각 검색엔진에서 반환하게 될 문서의 수를 조정하여 모든 검색엔진으로부터 120개의 결과를 가져온다.

● 방법3(엔진배제) : 적합도가 제일 낮은 검색엔진을 하나를 배제하고 검색엔진에서 가져올 반환문서의 수를 각 30개씩으로 정하여 총 90개의 결과를 가져온다.

● 방법4(배제&조정) : 적합도가 제일 낮은 검 하나를 배제하고 식(6)을 이용하여 반환하게 될 문 를 달리하여 검색엔진으로부터 총 90개의 결과를

<표 2> 질의에 대한 유용도

query		NL	AV	EX	IS	평균
1	game	42.24144	65.11322	6.83127	11.54398	31.43248
2	travel	50.82842	168.83277	11.60811	17.37782	62.16178
3	music	62.58327	243.83092	17.65927	20.83646	86.22748
4	sport	70.70417	80.09504	3.15049	14.11660	42.01657
5	yahoo	19.09245	65.72678	5.19965	2.92480	23.23592

<표 3> 반환시간

	NL	AV	EX	IS	평균
평균응답시간	0.36	0.93	0.35	0.65	0.5725
Presentation Time	2.522	1.569	2.558	1.877	2.1315

<표 4> 질의에 대한 적합도

		NL	AV	EX	IS	평균
1	game	44.76357	66.68241	9.38917	13.42096	33.56403
2	travel	53.35054	170.40196	14.16601	19.25481	64.29333
3	music	65.10539	245.40011	20.21717	22.71345	88.35903
4	sport	73.22629	81.66423	5.70839	15.99359	44.14812
5	yahoo	21.61458	67.29597	7.75755	4.80179	25.36747

가져온다.

● 방법5(m조정) : 검색엔진을 배제하지 않고 식(6)을 이용하여 검색엔진에서 반환하게 될 문서의 수를 조정하여 각 검색엔진으로부터 90개의 결과를 가져온다.

위의 다섯 가지 방법으로 각 질의에 대한 검색엔진의 선택 및 검색엔진에서 가지고 올 반환문서의 수를 <표 5> ~ <표 9>와 같이 결과조정 표를 만들었다. 사용자 질의에 대한 결과조정 표에 의해 메타서치엔진에 질의를 보내고 각 검색엔진에서 반환된 검색결과를 적합도 산정기준에 의하여 <표 10>과 같이 적합한 문서의 수를 구하였다.

다섯가지방법으로 각 질의를 메타서치엔진에서 검색한 결과는 방법1에서처럼 단순 적용한 경우보다 방법2처럼 검색엔진의 적합도를 구하여 가지고 올 반환문서의 수를 조정한 경우는 적합한 문서가 증가함을 볼 수 있다. 네트워크의 트래픽을 줄이기 위해서 메타서치엔진에서 가지고 올 총 반환문서의 수를 줄이는 방법에서는 단순히 검색엔진을 배제한 방법3보다는 검색엔진을 배제하면

<표 5> 질의1(game)에 대한 결과조정 표

실험방법		NL	AV	EX	IS	m
1	단순적용	30	30	30	30	120
2	결과조정	40	60	8	12	120
3	엔진배제	30	30	-	30	90
4	배제&조정	32	48	-	10	90
5	m 조정	30	45	6	9	90
적합도산정기준		게임에 대한 소개가 있으면 적합한 문서				

<표 6> 질의2(travel)에 대한 결과조정 표

실험방법		NL	AV	EX	IS	m
1	단순적용	30	30	30	30	120
2	결과조정	25	79	7	9	120
3	엔진배제	30	30	-	30	90
4	배제&조정	20	63		7	90
5	m 조정	19	59	5	7	90
적합도산정기준		여행에 관한 소개가 있으면 적합한 문서				

<표 7> 질의3(music)에 대한 결과조정 표

실험방법		NL	AV	EX	IS	m
1	단순적용	30	30	30	30	120
2	결과조정	22	83	7	8	120
3	엔진배제	30	30	-	30	90
4	배제&조정	18	66	-	6	90
5	m 조정	17	62	5	6	90
적합도산정기준		음악에 관한 소개가 있으면 적합한 문서				

<표 8> 질의4(sport)에 대한 결과조정 표

실험방법		NL	AV	EX	IS	m
1	단순적용	30	30	30	30	120
2	결과조정	50	55	4	11	120
3	엔진배제	30	30	-	30	90
4	배제&조정	39	43	-	8	90
5	m 조정	37	42	3	8	90
적합도산정기준		스포츠에 관한 소개가 있으면 적합한 문서				

<표 9> 질의5(yahoo)에 대한 결과조정 표

실험방법		NL	AV	EX	IS	m
1	단순적용	30	30	30	30	120
2	결과조정	25	80	9	6	120
3	엔진배제	30	30	30	-	90
4	배제&조정	20	63	7	-	90
5	m 조정	19	60	7	4	90
적합도산정기준		야후회사나 야후검색엔진에 대한 소개가 있으면 적합한 문서				

<표 10> 질의에 대한 적합한 문서의 수

실험방법		m (총결과수)	D_i (적합문서수)				
			q1	q2	q3	q4	q5
1	단순적용	120	103	98	100	104	100
2	결과조정	120	102	103	107	105	109
3	엔진배제	90	77	74	76	79	76
4	배제&조정	90	79	79	82	80	83
5	m 조정	90	81	80	82	81	83

<표 11> 질의에 대한 적합율

실험방법		m (총결과수)	Precision Ratio (적합율)				
			q1	q2	q3	q4	q5
1	단순적용	120	85.8%	81.7%	83.3%	86.7%	83.3%
2	결과조정	120	85.0%	85.8%	89.2%	87.5%	90.8%
3	엔진배제	90	85.6%	82.2%	84.4%	87.8%	84.4%
4	배제&조정	90	87.8%	87.8%	91.1%	88.9%	92.2%
5	m 조정	90	90.0%	88.9%	91.1%	90.0%	92.2%

서 검색엔진에서 가지고 올 반환문서의 수를 조정한 방법4에 적합한 문서가 많이 있음을 볼 수 있다. 그리고 검색엔진을 배제하지 않고 총 반환문서의 수를 줄이면서 각 검색엔진에서 가지고 올 문서의 수를 조정한 방법5에 적합한 문서의 개수가 제일 많이 검색되었다. <표 11>은 적합한 문서를 산정 하여 얻은 적합율의 결과를 나타낸다. 방법1의 단순 적용한 경우와 적합도 점수가 가장 낮은 검색엔진을 배제 한 방법3의 경우에서 검색엔진을 배제하여 네트워크의 트래픽을 줄여도 사용자 질의에 대한 적합율은 그대로 유지한다는 것을 알 수 있다.

다섯가지방법 중 방법5의 경우가 적합율이 제일 높은 것으로 나와 단순히 검색엔진을 배제하는 경우보다 검색엔진을 유지하면서 메타서치엔진에서 가지고온 총 문서의 수를 줄이고, 각 검색엔진에서 가지고 올 문서를 조정하는 것이 네트워크의 필요이상의 트래픽을 줄이고 검색효율을 높일 수 있었다.

본 논문에서는 사용자가 검색하는 모든 질의에 대한 우선 순위를 결정하여 기록을 유지하기가 어려워 대표적인 질의에 대한 유용도의 평균값을 이용하여 각 검색엔진의 적합도를 구하여 일정 기간 동안 사용하기 위하여 Search Engine Watch에서 제공하는 빈도수가 많은 질의 100개에 대한 인덱스를 만들고 인덱스를 이용하여 각 질의에 대한 검색엔진의 유용도를 계산하여 평균값을 구하였다. 사용자가 검색한 질의에 대한 검색엔진의 적합도는 질의에 대한 검색엔진의 유용도와 네트워크의 처리 속도를 나타내는 반환시간을 식(5)에 이용하여 <표 12>와 같이 구하였다. 적합도의 결과를 식(6)에 적용하여 검색엔진을 선택하거나

각 검색엔진에서 가지고 올 반환 문서의 수를 위에서와 같이 다섯 가지의 방법으로 결과조정 표를 만들었다

다섯 가지 방법으로 각 질의에 대한 검색엔진의 선택 및 검색엔진에서 가지고 올 문서의 수를 <표 13>과 같이

<표 12> 갱신주기 동안에 적용될 적합도

	NL	AV	EX	IS	합
Presentation Time	2.522	1.569	2.558	1.877	2.132
usefulness	10.73274	19.64959	4.03833	8.07666	10.62433
Fitness	13.25487	21.21878	6.59623	9.95364	12.75588

<표 13> 평균치에 대한 결과조정 표

실험방법		NL	AV	EX	IS	m
1	단순적용	30	30	30	30	120
2	결과조정	31	50	16	23	120
3	엔진배제	30	30	-	30	90
4	배제&조정	27	43	-	20	90
5	m 조정	23	37	12	18	90

결과조정 표를 만들고 메타서치엔진에 검색을 하였다. 결정된 다섯 가지 방법으로 검색엔진에 질의를 하고 그 결과의 적합성을 판정해 보았다. 한 개의 텀으로 된 질의로 <표 13>의 결과조정 표에 따라 질의한 경우의 결과를 식(7)에 적용하여 표 18과 같은 결과를 얻었다. 한 개 텀으로 질의한 결과인 <표 18>에서는 방법1의 단순적용 했을 경우 결과의 수 120개중에 평균 적합한 문서는 101.0개로 검색한 질의결과의 적합한 비율이 84.2%로 나왔으며 방법3에서와 같이 적합도 점수가 가장 낮은 검색엔진을 배제한 경우는 총 문서 90개중에 평균 적합한 문서가 76.4개로 84.9%로 방법1에서 단순 적용한 경우와 유사한 결과를 얻었다. 이는 유용도가 낮다고 해서 무조건 배제한 경우는 배제된 검색엔진의 결과 중에 앞쪽에 있는 적합성이 높은 문서까지도 배제되었기 때문이다.

방법2처럼 각 검색엔진에 구해진 평균 적합도를 식(6)에 적용하여 각 검색엔진마다 결과의 수를 달리한 경우는 총 120개 문서 중 평균 적합한 문서가 104.8개로 적합한 비율이 87.3%로 적합한 비율이 단순 적용한 실험 1의 방법보다 3.1% 적합한 비율이 높아졌다. 네트워크의 트래픽을 감소하기 위해서 방법4와 방법5처럼 총 결과의 수를 90으로 줄이고 적합도 점수가 가장 낮은 검색엔진을 배제하고 적합도를 식(6)에 적용하여 각 검색엔진마다 결과의 수를 달리한 경우는 90.0%로 단순 적용한 방법1보다 네트워크의 트래픽을 줄이면서도 적합한 비율은 5.8%로 증가하였으며 방법5처럼 검색엔진을 배제하지 않고서 검색엔진을 배제한 수만큼의 결과의 수를 적게 하고 적합도를 식(6)에 적용하여 각 검색엔진마다 결과의 수를 달리한 경우는 총 문서의 수 90개 중 평균 적합한 문서는 81.4개로서 적합한 비율이 90.4%로 단순 적용한 경우보다 6.2%가 증가하여 가장 적합한 문서의 비율이 높았다. 두 개의 텀으로 질의한 경우는 한 개의 텀으로 질의한 경우와 같은 결과를 얻을 수 있었으며 그 결과는 <표 15>에 나타난다.

두 개의 텀으로 질의한 경우는 한 개 질의한 경우 보다 전체적으로 적합한 비율이 조금씩 낮아지는 이유는 텀이 많아질수록 적합한 문서는 줄어들게 때문이다. 한 개의 텀으로 된 결과 <표 14>과 두개의 텀으로 얻어진 <표 15>를 합해서 하나의 종합결과를 만들면 <표 16>과 같은 결과 값을 얻게 된다. 검색엔진을 선택하여 가져올 문서의 수를 차등 적용하는 것이 동일한 수를 적용한 방법보다 검색결과의 적합율이 평균 4.9%가 높아졌다. 검색엔진을 배제하는 선택방법은 질의에 대한 적합율을 유지하면서 네트워크상의 트래픽을 감소하였으나 검색엔진을 배제하지 않고 총 결과의 수만을 감소하였을 때는 네트워크의 트래픽

<표 14> 한 개 텀으로 검색한 결과

실험방법		m (총결과수)	Avg D_i (적합문서수)	Precision Ratio (적합율)
1	단순적용	120	101.0	84.2%
2	결과조정	120	104.8	87.3%
3	엔진배제	90	76.4	84.9%
4	배제&조정	90	81.0	90.0%
5	m 조정	90	81.4	90.4%

<표 15> 두개 텀으로 검색한 결과

실험방법		m (총결과수)	Avg D_i (적합문서수)	Precision Ratio (적합율)
1	단순적용	120	94.7	78.9%
2	결과조정	120	102.7	85.5%
3	엔진배제	90	71.2	79.1%
4	배제&조정	90	80.9	89.9%
5	m 조정	90	81.8	90.8%

<표 16> 종합결과

실험방법		m (총결과수)	Avg D_i (적합문서수)	Precision Ratio (적합율)
1	단순적용	120	97.8	81.5%
2	결과조정	120	103.7	86.4%
3	엔진배제	90	73.8	82.0%
4	배제&조정	90	81.0	89.9%
5	m 조정	90	81.6	90.6%

은 감소시키면서 적합율은 평균 9.1%가 높아졌다.

4. 결 론

본 논문에서는 검색엔진의 평가를 통하여 선택된 검색엔진이나 문서의 결과의 수를 조정하여 정보검색을 수행하게 함으로써, 검색된 결과의 사용자 만족도를 유지하면서 네트워크 상의 불필요한 트래픽을 감소시키는 방안을 제안하였다. 질의어에 대한 검색엔진의 유사도의 값은 실험한 검색엔진 중 가장 많은 웹 페이지를 보유하고있는 Northern Light는 평균치를 유지하였으며 AltaVista는 평균치보다 9.0정도의 높은 값을 보였고 Excite는 평균치보다 6.3이나 작았다. 검색된 결과에 의하면 AltaVista가 검색결과에 대한 효율이 제일 좋았고 Excite와 InfoSeek는 전반적으로 불필요한 문서가 존재하였으나 상위 10개 문서에 대하여는 효율이 높은 것을 보았다. 그러므로 질의에 대한 검색엔진의 유사도가 낮은 검색엔진도 배제하는 것보다는 반환할 문서의 수를 조정하는 것이 사용자의 만족도를 높일 수 있었다.

참 고 문 헌

[1] 김화수,"정보검색포럼,"
http://www.igate.co.kr/review/isf/seca1.html1998.

[2]정태충, "Meta-Search Engine에 기반한 인터넷 검색 에이전트 개발에 관한 연구," 경희대학교 전자계산공학과 석사논문, 1997.

[3]한국정보통신진흥협회, "인터넷 정보검색사," 박문각, 1998.

[4] Clement Yu, "Metasearch Engines :Solution and Challenges tutorial," Proceedings of the 25th International Conference Very Large Database, 1999.

[5] Weiyi Meng, King-Lup Liu, Clement Yu, Wensheng Wu, and Naphtali Rishe, "Estimating the Usefulness of Search Engines," IEEE International Conference on Data Engineering, 1999.

[6] Budi YuWono, Dik Lee, "Server Ranking for Distributed Text Resource System on the Internet," Proceedings of the Fifth International Conference On Database System for Advanced Applications, 1997.

[7]고형일, "Web 상에서의 정보 검색을 위한 지능형 에이전트의 설계 및 구현," 금오공과대학교 전자공학과 석사학위논문, 1997.

[8] King-Lup Liu, Clement Yu, Weiyi Meng, Wensheng Wu, Naphtali Rishe "A Statistical Method for Estimating the Usefulness of Text Database," IEEE Transaction on Knowledge and Data Engineering, 1999.

[9] Weiyi Meng, King-Lup Liu, Clement Yu, Xiaodong Wang, Naphtali Rishe, "Determining Text Database to Search in the Internet," Proceedings of the VLDB Conference New York, 1998.

이 진 호

1974년 영남대학교 전자공학과 졸업(공학사)
1981년 영남대학교 대학원 전자공학과 계산기 전공(공학석사)
1996년 영남대학교 대학원 전자공학과 전산공학 전공(공학박사)
1979년~현재 경일대학교 공과대학 컴퓨터공학과 교수
관심분야 : 데이터 압축, 프로그래밍 언어, 객체지향 시스템

박 상 호

1979년 경북대학교 전자공학과 (공학사)
1981년 영남대학교 대학원 전자공학과(공학석사)
1989년 Syracuse University(M.S.)
1995년 State University of New York at Buffalo(Ph.D.)
1996년~현재 안동대학교 정보통신공학과 조교수
관심분야 : 멀티미디어통신, 이동통신

박 선 진

1984년 2월 인천기능대학
1988년 창원기능대학
1998년 2월 상주대학교 컴퓨터공학과
2000년 2월 안동대학교 대학원 컴퓨터공학과
1992년 3월~현재 영주직업전문학교 근무
관심분야 : 정보검색, 전자상거래

남 인 길

1978년 경북대학교 전자공학과 (공학사)
1981년 영남대학교 대학원 전자공학과계산기전공(공학석사)
1992년 경북대학교 대학원 전자공학과 전산공학전공(공학박사)
1978년~1980년 대구은행 전산부
1980년~1990년 경북산업대학 전자계산학과 부교수
1990년~현재 대구대학교 컴퓨터정보공학부 교수
1996년~1997년 미국 루이지애나 주립대학 교환교수
관심분야 : 데이터베이스, GIS, 이동컴퓨팅