Proceedings of KFIS Spring Conference 2007 Vol. 17, No. 1.

웹문서 재배치 에이전트 시스템[1)]

A Web Page Reordering Agent System

조영임 · 강상길 · 김영국

수원대학교 IT대학 컴퓨터학과 · 인하대학교 컴퓨터공학부 · 충남대학교 컴퓨터학과
E-mail: ycho@suwon.ac.kr · sgkang@inha.ac.kr · ykim@chungnam.ac.kr

요 약

구글이나 야후와 같은 정보검색엔진은 사용자에게 편리성을 제공하나 사용자로 하여금 만족감을 제공하지는 못하고 있다. 이것은 사용자에 대한 검색목표가 사용자 프로파일마다 서로 다르기 때문이다. 따라서 검색엔진으로 검색된 결과를 사용자 프로파일에 따라서 재배치하는 것은 매우 필요하다. 이 논문에서는 키워드기반 검색엔진으로 검색된 결과를 사용자 프로파일에 따라 웹문서를 재배치하는 알고리즘을 제안한다. 각 키워드에 대한 가중치는 사용자가 웹문서에 대해서 수행한 행동 즉, 다운로드, 클릭, 아무행동 안함에 따라 차등 적용하여 업데이트하여 웹문서를 리스트하여 사용자에게 제공한다.

Key Words : Web pages reordering, Agents, Information search engine

1. 서 론

인터넷의 정보의 홍수속에서 사용자들은 정보검색을 통해 많은 정보를 접하게 된다. 구글이나 야후와 같은 정보검색엔진을 통한 검색으로부터 사용자들에게 맞는 정보를 찾기 위한 시간과 노력 줄이기 위해서는 웹 문서 리스트가 필요하다. 만약 쿼리가 동일한 검색엔진에 적용되더라도 사용자의 프로파일에 따라서 다르게 사용되어야 한다. 그러나 기존 검색엔진은 모든 사용자에 대해 동일한 결과를 보여주기 때문에 사용자 프로파일에 따른 웹 페이지 재배치 방법이 필요하다.

본 논문에서는 이러한 목적에 의해, 사용자 프로파일에 따라서 웹 문서를 재배치하는 알고리즘을 제안하고자 한다. 기존에 Page 등에 의해 제안된 방법들이 있으나 웹의 그래프구조에서의 웹 문서 위치에 따라서 랭크하는 방법이 있었다[ ]. 또한 Weiss 등도 하이퍼텍스트 구조에서 계층적 검색엔진방법을 제안하였다[ ]. 그러나 이들방법은 사용자의 프로파일을 근거로 제안하지 않고 검색엔진에서의 구조를 변경하는 방법을 제안한 것이다.

본 논문에서 제안하는 알고리즘은 사용자의 검색엔진에서의 쿼리 히스토리를 바탕으로 사용자의 개인 키워드 데이터베이스를 구축한다. 키워드는 전치사와 관사를 제외한 명사로 제한한다. 만약 어떤 단어가 웹 문서에서 임의의 쓰레숄드 이상 사용된다면 이를 키워드로 간주하여 데이터베이스에 저장한다. 그리고, 그 키워드를 포함하는 웹 문서의 검색결과에서의 랭킹에 따라 키워드에 가중치를 부여한다. 본 논문에서 제안하는 시스템은 각 웹 문서에 포함된 키워드의 전체 가중치에 의해 거색엔진으로부터 검색된 웹 문서를 재배치한다. 또한 각 키워드 가중치는 사용자의 행동 즉, 다운로드, 클릭, 행동안함 등과 같은 사용자의 행동에 의해 업데이트된다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 제안하는 시스템을 설명하고 3장에서 시스템 성능평가를 하고 4장에서 결론을 맺고자 한다.

2. 제안하는 시스템

다음 그림 1은 본 논문에서 제안하는 전체 시스템 구조를 나타낸 것이다. 만약 사용자가 검색엔진에 키워드를 입력함으로써 쿼리를 하면, 검색엔진은 쿼리와 관련된 웹 문서를 리스

1) 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원 사업(IITA-2006-C1090-0603-0031)의 연구결과로 수행되었음

트하여 보여주게 된다.

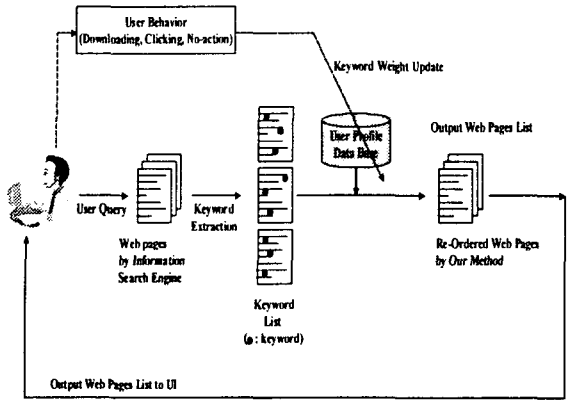

그림 1. 제안하는 전체 시스템 구조

키워드는 웹 문서로부터 추출될 수 있다. 앞의 연구로부터 키워드가 출현하는 빈도수로 측정될 수 있다면, 일정한 쓰레숄드 값 이상을 갖는다면 키워드로 간주할 수 있다.

키워드는 키워드 인덱스로 구성된 키워드 데이터베이스를 구성한다. 여기서 i 번째 키워드를 k_i로 나타내고 가중치는 w_i로 나타낸다.

키워드 가중치 w_i 는 초기에는 검색엔진에서 검색된 웹 문서 리스트에 있는 키워드 가중치에 의해 초기화된다. 그러나 키워드는 그림 2에서와 같이 다중으로 웹 문서에 나타날 수 있다. 그래서 다음 식(1)과 같이 키워드 데이터베이스에 표현된다. 여기서 N 은 검색된 웹 문서 리스트에 있는 웹 문서의 총 개수를 말하고, $R_{i,j}$ $(j \in N)$ 는 키워드 i를 포함하는 j 번째 웹 문서의 검색 랭크를 말한다. 다음 식(1)에 의해서 w_i는 높에 랭크된 문서일수록 점차적으로 키워드 가중치가 증가하게 된다.

$$w_i = \sum_j \frac{N - R_{i,j} + 1}{N} \qquad (1)$$

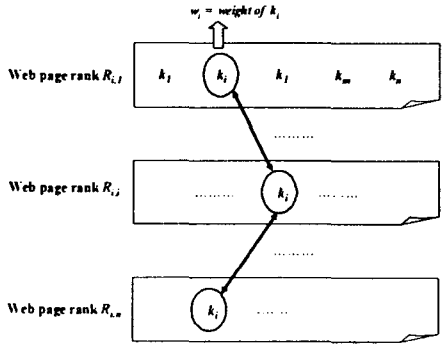

그림 2. 다중 출현하는 키워드의 가중치

정규화된 가중치를 이용하여 다음 식(2)와 같이 j번째 각 웹 문서의 전체 관련성 Rev_j 을 계산할 수 있다. 여기서 K_j는 j번째 웹 문서에 포함된 키워드 집합을 의미한다.

$$\mathrm{Re}v_j = \sum_{k_i \in K_j} \tilde{w}_i \qquad (2)$$

또한, 정규화된 가중치 $\tilde{w}_i$는 다음 식(3)과 같이 키워드에 대한 사용자의 행동들 u (다운로드, 클릭, 행동안함)에 의해 가중치가 업데이트된다.

$$\tilde{w}_i = \tilde{w}_i \cdot (1+u) \qquad (3)$$

3. 성능 평가

본 논문에서 제안하는 시스템의 성능평가를 위해 윈도우 환경에서 비주얼 C++을 이용하여 구현하였고 구글에서 검색된 결과를 일주일동안 분석하여 실험하였다.

실험조건은 다음과 같다. 본 논문에서는 명사만을 키워드로 간주하였다. 또한 쓰레숄드는 각 웹 문서에서 5번 이상 출현하는 키워드를 키워드 데이터베이스에 넣어 구성하였다.

다음 그림 3은 구글에서 'genetic'을 키워드로 입력했을 때 검색된 화면이다. 이때 890개의 웹 문서가 검색되었다.

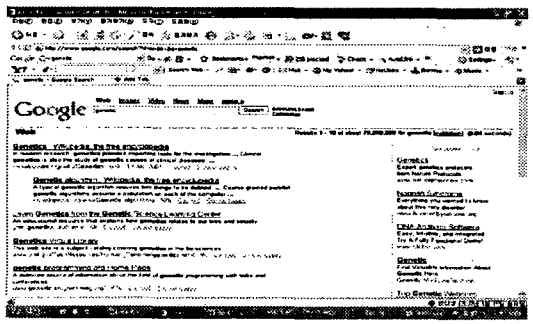

그림 3. 구글에서 'genetic' 입력시 검색결과

이러한 구글의 검색결과로부터 2, 5, 25번째로 랭크된 웹 문서로부터 추출된 키워드 집합 K_2, K_5, K_{25}는 다음 그림 4와 같다. 예를 들면 그림에서 보는바와 같이 'genetic', 'algorithm', 'engineering', 'genome', 'project', 'programming', 'neuron' 등과 같은 키워드들이 2번째로 랭크된 웹 문서로부터 추출된 키워드들이다. 다른 키워드 집합들도 같은 방법으로 추출되어 구성한다.

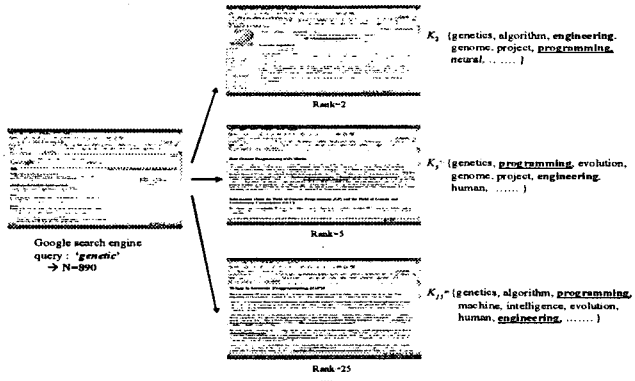

그림 4. 2,5,25위로 랭크된 웹 문서들

또한 실험을 통해 일주일동안 구글에서 검색된 문서들로부터 구축된 사용자 프로파일 데이터베이스 구조는 다음 표 1과 같다.

표 1. 사용자 프로파일 데이터베이스

1	algorithm	0.678		...	
	...		179	human	0.204
13	biology	0.218		...	
	...		195	information	0.245
34	chromosome	0.391	196	inheritance	0.224
	...			...	
56	circuit	0.209	215	instance	0.098
	...		216	intelligence	1.000
67	cluster	0.321		...	
	...		225	invention	0.196
77	code	0.798		...	
	...		285	machine	0.316
89	computation	0.324		...	
90	computer	0.503	332	network	0.168
	...			...	
108	DNA	0.698	344	nucleotide	0.119
	...			...	
115	field	0.228	402	organism	0.298
	...			...	
152	gene	0.897	413	pattern	0.723
153	genetics	0.912		...	
154	genome	0.542	421	population	0.121

Index	Keyword	Weight
	...	
429	problem	0.211
435	programming	0.876
	...	
444	protein	0.112
	...	
457	result	0.354
	...	
463	RNA	0.191
	...	
491	selection	0.252
	...	
510	sequence	0.415
	...	
517	statement	0.427
	...	
525	structure	0.504
	...	
541	synthesis	0.456
	...	
569	technique	0.452
	...	

다음 그림 5는 그림 3의 구글 검색결과를 본 논문에서의 웹 문서를 재배치 알고리즘에 의해 재배치하여 사용자에게 보여준 결과이다. 이 그림에서 5번째로 랭크된 웹 문서의 Rev_5 는 9.158(=0.678+0.206+0.321+0.324+0.503+0.228+0.897+0.912+0.234+0.245+0.098+1+0.316+0.168+0.211+0.876+0.354+0.252+0.427+0.456+0.452)이므로 본 시스템에서는 1위로 재배치되었고, 반면 구글에서 1위로 랭크된 문서의 Rev_1 는 7.36이 되어 5위로 랭크되도록 재배치되었다.

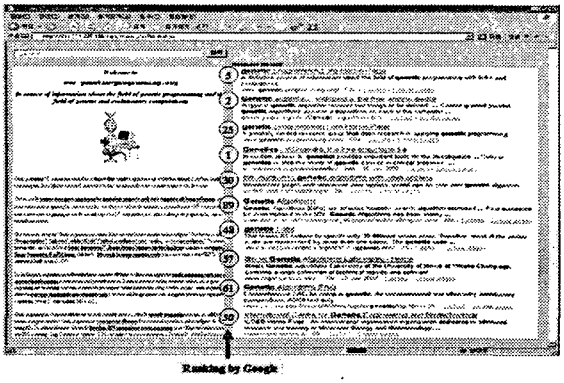

그림 5. 재배치된 웹 문서

이에 따라서 표 1의 사용자 프로파일 데이터베이스도 자동적으로 업데이트하게 된다. 만약 사용자가 아무행동을 안하면 u=0, 클릭을 하면 u=0.1, 다운로드하면 u=0.2로 되어서, 사용자 프로파일 데이터베이스에서 관련된 키워드들이 자동적으로 업데이트된다.

이 시스템의 성능평가를 위해 17명의 대학원생으로부터 약 1주일동안 성능평가해본 결과, 본 논문에서 제안한 시스템으로부터 재배치되어 검색된 결과의 상위 5위안의 문서들은 사용자 클릭 빈도수가 매우 높았다. 이는 구글과 비교해보면, 본 논문의 재배치 결과가 매우 우수함을 알 수 있다. 즉, 사용자들은 상위 5위안의 웹 문서들에 매우 관심이 높음을 알 수 있다. 그러나 30위 이하의 웹 문서부터는 구글이나 본 시스템이나 별 차이가 없음을 알 수 있다.

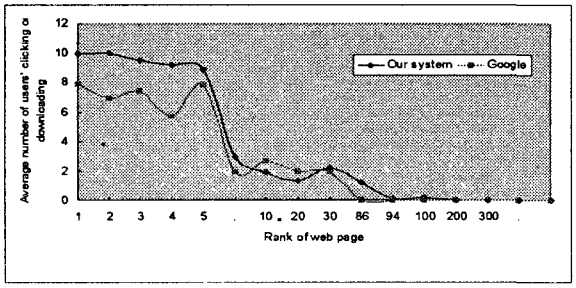

그림 6. 평균 클릭횟수(구글 vs. 본 시스템)

4. 결론

본 논문에서는 기존의 정보검색엔진으로부터 검색된 웹 문서를 사용자 프로파일을 기준으로 재배치하여 사용자의 만족도를 높이는 시스템을 제안하였다. 실험결과 구글보다는 본 시스템에서 사용자 만족도가 매우 높았으며 특히 상위 5개 문서에 대한 사용자 만족도가 기존 검색엔진보다 매우 높았다. 이 논문은 앞으로 맞춤형 정보검색에 응용하면 효과적일 것이다.

앞으로 본 논문에서 사용된 가중치 변화정도, 쓰레숄도값 등이 좀더 많은 실험을 통해 최적화되어야 할 것이다.

참 고 문 헌

[1] Page, L., Brin, S., Motwani, R., Winograd, T.: ThePageRank Citation Ranking: Bring Order to the Web. Stanford Digital Library Technologies Project. 1998

[2] Brin, S., Page, L.: The Anatomy of a Large-scale Hypertexture Web Search Engine. Proc. The 7th International Conference on World Wide Web 7, pp.107-117, 1998

[3] Weiss, R., Vélez, B., Sheldon, M.A., Manprempre, C., Szilagyi, P., Duda, A., Gifford, D.K.: HyPursuit: A Hierarchical Network Search Engine that Exploits Content-link Hypertext Clustering. Proc. The 7th ACM Conference on Hypertest, pp.180-193, 1996

[4] Baeza-Yates, R., Castillo, C.: Relating Web Characteristics with Link based Web Page Ranking. String Processing and Information Retrieval, pp. 21-32, 2001

[5] Baeza-Yates, R., Davis, E.: Web Page Ranking using Link Attributes. International World Wide Web Conference, pp. 328-329, 2004

[6] Hristidis, V., Papakonstantinou, Y.: DISCOVER: Keyword Search in Relational Databases. Proc. VLDB Conference, 2002

[7] Hristidis, V., Papakonstantinou, Y., Balmin, A.: Keyword Proximity Search on XML Graphs. Proc. International Conference on Data Engineering, pp. 367-378, 2003

[8] Agrawal, S., Chaudhuri, S., Das, G.: DBXplorer: A System for Keyword-based Search over Relational Databases. Proc. International Conference on Data Engineering, pp. 5-16, 2002

[9] Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., Sudarshan, S.: Keyword Searching and Browsing in Database using BANKS. Proc. International Conference on Data Engineering, pp. 5-16, 2002

[10] Xu, Y., Papakonstantinou, Y.: Efficient Keyword Search for Smallest LCAs in XML Database. Proc. The 2005 ACM SIGMOD International Conference on Management of Data, pp. 527-538, 2005

[11] Guo, L., Shao, F., Botev, C., Shanmugasundaram: XRANK: Ranked Keyword Search over XML Documents. SIGMOD, 2003

[12] Kang, S., Cho, Y.: A Novel Personalized Paper Search System. Lecture Notes in Computer Science, Vol. 4113, pp. 1257-1262, 2006