퍼지 개념 네트워크를 이용한 링크기반 검색엔진의 개인화

김경중[0] 조성배
연 세 대 학 교 컴 퓨 터 과 학 과
uribyul@candy.yonsei.ac.kr sbcho@csai.yonsei.ac.kr

Personalization of Link-based Search Engine by Fuzzy Concept Network

Kyung-Joong Kim[0] Sung-Bae Cho
Dept. of Computer Science, Yonsei University

요 약

링크 기반 검색엔진은 사용자의 질의어와 관련된 웹문서들에 대해 링크 정보를 이용하여 순위를 생성한다. 링크 정보는 문서들간의 추천을 나타내므로 중요한 문서를 찾는데 이용할 수 있다. 링크 정보를 이용한 검색은 일반적인 텍스트 기반 검색엔진에 비해 좋은 결과를 낸다고 알려져 있다. 링크 정보를 이용한 검색엔진의 대표적인 예로는 Google과 Clever Search가 있다. 본 논문에서는 링크 정보를 이용한 검색엔진을 개발하고 퍼지 개념 네트워크를 이용하여 개인화를 수행한다. 퍼지 개념 네트워크는 퍼지 문서 추출 시스템을 위한 지식베이스로 이용된다. 사용자 프로파일을 이용하여 사용자별로 퍼지 개념 네트워크를 생성하고 링크 기반 검색 결과를 개인화한다. 3명의 사용자에 대해 실험을 수행하여, 개인화가 주는 효과에 대해 평가한다. 검색결과는 중요한 웹 문서를 찾아 주었으며, 개인화 과정을 통해 사용자가 원하는 순서대로 정렬해 주는 것을 알 수 있었다.

1. 서론

링크 정보를 이용한 웹 검색엔진은 최근들어 인기를 얻고 있다. 대표적인 예로 Stanford 대학에서 개발한 Google이 있다[1]. Google은 빠른 검색속도와 정확한 검색결과로 대표적인 검색엔진중의 하나로서 인정받고 있다. Google이외에도 IBM이 개발중인 Clever Search도 링크 정보를 이용하여 보다 나은 검색결과를 제공하려고 개발되고 있다[2].

링크 정보를 이용한 정보추출은 과학 인용도 조사와 비슷한 개념이다. 즉, 저명한 저널이 참조하고 있는 저널이 높은 순위를 받게 되는 것처럼 신뢰도 높은 웹 문서로부터 링크받고 있는 웹 문서가 높은 순위를 받는 것이다. PageRank기법이 이와 같은 방법을 이용하여 웹 문서의 순위를 평가하며, Google에서 이용하고 있다[1].

링크 정보를 이용한 검색엔진은 텍스트 기반 검색엔진의 한계점을 해결해줄 것으로 기대를 모으고 있다. 현재의 검색엔진은 텍스트 정보만을 이용하기 때문에 검색결과를 향상시키는데 한계를 지니고 있다. 또한, 텍스트 정보를 이용한 의도적인 순위높이기에 취약한 단점이 있다.

본 논문에서는 링크 정보를 이용하여 검색을 수행하는 시스템을 개발하고, 개인화를 위해 퍼지 개념 네트워크를 사용한다. 또한, 3명의 사용자에 대해 테스트를 수행하여 성능을 평가해 본다.

2. 관련연구

링크 정보를 이용한 검색엔진을 개발한 사례로 Google과 Clever Search가 있다. Google은 Stanford 대학에서 개발한 링크기반 검색엔진으로 현재는 상용화되어 서비스되고 있다. Google은 PageRank를 이용하여 웹문서의 중요도를 사전에 계산해 놓고, 사용자의 질의어가 들어오면 관련 웹 문서를 찾아내고 PageRank값을 기준으로 순위를 생성하여 결과를 보여준다[1].

Clever Search는 IBM이 개발하고 있는 지능형 검색엔진이다[2]. 이것은 링크 정보를 이용한 웹 문서 순위 생성 알고리즘을 적용하여 웹 문서중에서 가장 신뢰도가 높은 authoritative문서와 많은 authoritative문서를 링크하고 있는 hub문서를 가려준다. 예를 들어 "java"에 관한 authoritative문서로는 "java.sun.com"을 들 수 있다. 그리고 대표적인 hub 사이트로는 "Yahoo!"를 들 수 있다. Clever Search의 장점은 두가지 형태의 유용한 결과를 제공한다는 점이다. Clever Search 연구팀은 링크정보를 이용하여 웹 Community를 추출하거나 웹 문서를 클러스터링하는 작업도 함께 연구중이다. Clever 검색엔진은 Yahoo와 같은 디렉토리 서비스를 자동적으로 구축하는데 이용될 수 있다.

3. 링크 기반 검색엔진

링크기반 검색엔진은 크게 웹 문서로부터 링크 정보를 추출하는 부분, 링크 정보를 이용하여 검색을 수행하는 부분, 개인화를 수행하는 부분으로 나뉜다. 그림 1은 링크 기반 검색엔진의 구조도를 보여준다.

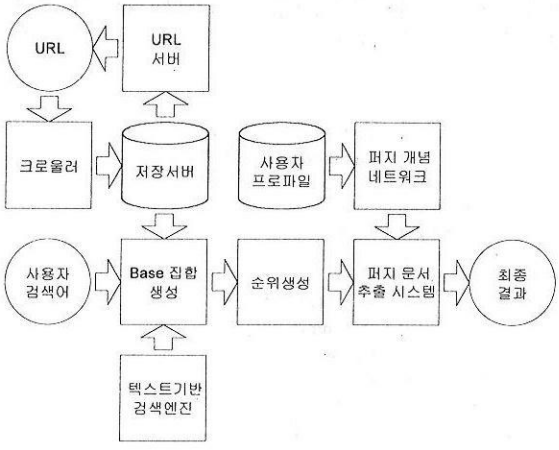
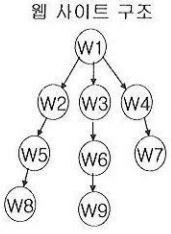


그림 1. 링크 기반 검색엔진의 구조도

3.1 링크 정보의 저장

웹 문서를 가져오고 링크 정보를 추출하는 작업은 크로울러가 담당한다. URL서버는 저장서버로부터 추출할 URL을 가져와 크로울러에게 전달한다. 크로울러는 해당 URL의 문서를 가져와 링크를 추출하게 된다. 추출된 링크는 저장서버를 통해 저장된다. 저장서버는 링크정보와 URL정보를 관리하며, 중복되는 것을 방지한다. URL정보는 유일한 *DocID*로 표현되며, 링크정보는 <*DocID*,*DocID*>의 형태로 저장된다. 그림 2는 링크정보가 저장되는 과정을 보여준다.



URL 정보

DocID	URL
00001	W1
00002	W2
00003	W3
00004	W4
00005	W5
00006	W6
00007	W7
00008	W8
00009	W9

Link 정보

DocID	DocID
00001	00002
00001	00003
00001	00004
00002	00005
00003	00006
00004	00007
00005	00008
00006	00009

그림 2. 링크 정보의 저장과정

3.2 순위 생성

사용자의 질의가 들어오면 base집합을 구성하고 링크정보를 이용하여 순위를 생성한다. 순위를 생성하기 위해 저장서버로부터 문서들간의 링크관계에 대한 정보를 얻어온다.

사용자의 질의어와 관련된 100개의 문서를 일반검색엔진으로부터 얻어온후, 이들이 링크하고 있는 문서와 이들을 링크하고 있는 문서를 포함하여 base집합을 형성한다. 초기 100개의 웹 문서에는 포함되지 못했던 중요한 웹 문서들도 확장과정을 통해 base집합에 포함된다.

Base집합에 포함된 문서 i에 대해 authoritative 가중치 a_i와 hub가중치 h_i를 부여한후 1.0으로 초기화한다.

$$a_i = \sum h_j \text{ (문서 } j \text{는 문서 } i \text{를 링크한다.)}$$

$$h_i = \sum a_j \text{ (문서 } i \text{는 문서 } j \text{를 링크한다.)}$$

각 문서의 가중치는 위 수식을 이용하여 갱신된다. 좋은 authoritative문서는 좋은 hub로부터 링크되고, 좋은 hub문서는 좋은 authoritative문서를 링크한다[3]. 반복은 수렴할 때까지 수행하며, 경험적인 실험을 통해 5번의 반복을 기준으로 하였다.

3.3 개인화

링크정보기반 검색엔진을 통해 생성된 결과를 이용하여 사용자에게 적합한 문서를 찾아주는 개인화를 수행한다. 사용자 프로파일은 개인화를 위한 기초 정보를 제공하며, 퍼지 개념 네트워크를 생성하는데 이용된다. 퍼지 개념 네트워크는 사용자의 개념을 표시하며 퍼지 문서 추출을 수행할 때 지식 베이스로서 이용된다.

퍼지 개념 네트워크는 노드들과 방향성 있는 링크로 구성되어져 있다[4]. 각각의 노드는 개념 또는 문서를 나타낸다. 각각의 방향성 있는 링크는 두 개념을 연결시키거나 개념과 문서 사이의 관계를 정의한다.

$$C = \{C_1, C_2, \cdots, C_n\}$$

C는 개념들의 집합을 나타낸다. 개념 C_i에서 C_j까지의 중요도가 α이고 개념 C_j에서 C_k까지의 중요도가 β일때, 개념 C_i에서 C_k까지의 중요도는 α와 β중에서 작은 값이 된다. 만약 C_i에서 C_k까지의 경로가 여러 개 일때는 가장 큰 값을 중요도로 선택한다. 문서는 $d_1, d_2, d_3, \cdots, d_n$으로 표시되며, 각 개념 $C_1, C_2, C_3, \cdots, C_n$에 대한 중요도를 계산하여 문서 디스크립터로 표현된다. 그림 3은 퍼지 개념 네트워크의 예를 보여준다.

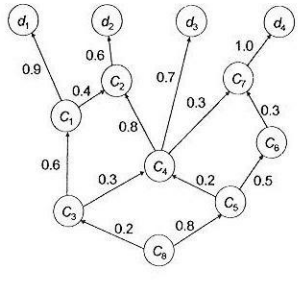


그림 3. 퍼지 개념 네트워크

퍼지 개념 네트워크를 이용하여 모든 문서의 중요도를 평가하는 작업은 시간이 오래 걸리기 때문에, 행렬형태로 표현된 퍼지 개념 행렬과 문서 디스크립터를 이용하여 퍼지 문서 추출을 수행한다. 퍼지 개념 행렬 K는 개념들 사이의 중요도를 원소값으로 가지고 있다. 문서 디스크립터 D는 문서가 $d_1, d_2, d_3, \cdots, d_m$으로 표시되며, 개념이 n개일때 아래와 같이 정의된다.($m \times n$ 행렬)

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix}$$

개념들 사이의 중요도가 모두 표현되어 있지 않기 때문에 전이폐쇄를 계산하여, 모든 개념들 사이의 중요도를 결정한다. 퍼지 개념 행렬의 곱은 퍼지 논리를 이용한다. $K^2 = K \otimes K$ 는 퍼지 개념 행렬 K의 곱을 표시한다. K는 n개의 개념을 표현하며, $n \times n$ 행렬이다.

$$K_{ij}^2 = \bigvee_{l=1}^{n} (K_{il} \wedge K_{lj}) \quad 1 \le i, j \le n$$

여기서 $\vee$와 $\wedge$는 각각 max연산과 min연산을 나타낸다. $K^p = K^{p+1} = K^{p+2} = \ldots$ 를 만족하며 $p \le n-1$인 정수 p가 존재하게 된다. $K^* = K^p$ 라고 하면 K^*는 퍼지 개념 행렬 K의 전이폐쇄가 된다. 각 개념에 대한 문서의 중요도는 문서 디스크립터 행렬 D와 퍼지 개념 행렬 K의 곱을 계산함으로써 향상될 수 있다[5].

$$D^* = D \otimes K^*$$

D^*를 확장된 문서 디스크립터 행렬이라고 부른다.

사용자가 질의어를 던지면 링크 정보기반 검색엔진은 5개의 authoritative 문서를 결정한다. 사용자 프로파일을 기초로 퍼지 개념 행렬을 생성하고, 5개의 문서에 대한 문서 디스크립터 행렬을 계산한다. 문서 디스크립터는 문서에 포함되어져 있는 개념의 빈도를 이용하여 평가한다. 사용자 프로파일에 사용된 n개의 개념에 대해 각 문서마다 빈도를 측정하고 0에서 1사이의 값으로 정규화한다. 문서 디스크립터 행렬과 퍼지 개념 행렬의 전이폐쇄를 곱하여 확장된 문서 디스크립터를 구한다. 확장된 문서 디스크립터에서 n개 개념에 대한 문서의 중요도를 모두 합하여 나온 값에 따라 순위를 사용자별로 다시 생성한다. 그림 4는 사용자 프로파일 정보를 기초로 퍼지 개념 행렬을 생성한 것을 보여준다.

사용자 프로파일

Java	Book	0.7
Java	Car	0.3
Java	WWW	0.9
Java	Ship	0.1
Book	Car	0.3
Book	WWW	0.5
Book	Ship	0.1
Book	Cafe	0.4
Car	WWW	0.7
Car	Ship	0.6
WWW	Ship	0.5
Ship	Cafe	0.3

퍼지 개념 행렬

	Java	Book	Car	WWW	Ship	Cafe
Java	1.0	0.7	0.3	0.9	0.1	0.0
Book	0.7	1.0	0.3	0.5	0.1	0.4
Car	0.3	0.3	1.0	0.7	0.6	0.0
WWW	0.9	0.5	0.7	1.0	0.5	0.0
Ship	0.1	0.1	0.6	0.5	1.0	0.3
Cafe	0.0	0.4	0.0	0.0	0.3	1.0

그림 4. 사용자 프로파일과 퍼지 개념 행렬

4. 실험결과

표 1은 링크 정보 기반 검색엔진이 선택한 "Java"에 대한 상위 5개의 authoritative문서와 hub문서들이다. 검색결과 "Java"에 대해 가장 신뢰를 받고 있는 "java.sun.com"이 authoritative문서중 1위로 선정되었다. 표 2는 개인화 결과를 보여준다. 개인화 결과 사용자 2의 경우 1위와 2위를 사용자가 원하는 문서로 찾아주었다.

표 1. "Java"에 대한 검색결과

Authoritative 결과	
1.	java.sun.com
2.	www.javalobby.org
3.	javaboutique.internet.com
4.	java.about.com/compute/java/mbody.htm
5.	www.javaworld.com
Hub 결과	
1.	industry.java.sun.com/products
2.	java.sun.com/industry
3.	java.sun.com/casestudies
4.	industry.java.sun.com/javanews/developer
5.	industry.java.sun.com/jug

표 2. 개인화 결과 (음영부분이 사용자가 결정한 순위와 일치)

사용자 1	사용자 2	사용자 3
2	1	2
1	2	1
3	3	3
4	4	5
5	5	4

5. 결론

본 논문에서는 링크 정보를 이용하여 중요한 웹 문서를 찾아내고 퍼지 문서 추출시스템의 지식베이스로 이용되는 퍼지 개념 네트워크를 사용자별로 구축하여 개인화를 수행하였다. 실험결과 중요한 문서를 찾아냈으며, 개인화에 대한 가능성을 확인했다.

6. 참고문헌

[1]S. Brin and L. Page," The anatomy of a large-scale hypertextual web serch engine," *The Seventh International WWW Conference*, 1998.

[2]The Clever Project, http://www.almaden.ibm.com/cs/k53/clever.html

[3]J. Kleinberg," Authoritative sources in a hyperlinked environment," *IBM Research Report RJ 10076*, 1997.

[4]S.-M. Chen and J.-Y. Wang," Document retrieval using knowledge-based fuzzy information retrieval techniques," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 5, pp. 793-803, 1995.

[5]C.-S. Chang and A.L.P. Chen," Supporting conceptual and neighborhood queries on the world wide web," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 2, pp. 300-308, 1998.