XLinks를 이용한 하이퍼텍스트 검색 시스템

김 은 정† · 배 종 민††

요 약

일반적인 하이퍼텍스트 검색 모델은 문서와 문서사이의 관계나 링크의 의미를 무시하고, 모든 문서를 독립적인 존재로 간주하여 검색한다. 그러나 하이퍼텍스트 검색 시스템에 있어 링크 정보를 이용하면 검색의 성능을 향상시킬 수 있다. 기존의 링크 기반 하이퍼텍스트 검색 모델은 문서의 색인 과정에서는 링크 정보를 무시하고, 검색 결과 집합에 대하여 문서의 우선 순위를 재조정하는데 링크 정보를 활용한다. 이는 링크 정보의 활용이 검색 결과 집합의 문서들에만 한정된다는 단점이 있다. 본 논문에서는 링크 정보를 문서의 색인 과정에서 활용한다. 색인 과정에서 링크 정보를 이용하여 문서 내 용어의 가중치와 문서 내 inLinks의 가중치를 정의하고, 이들을 이용하여 문서의 우선 순위를 위한 확장된 RSV 계산식을 제시한다. 실험 결과에서 링크 의미에 따른 검색 조회율과 정확도를 제시하고 기존 링크 기반 검색 모델과의 비교, 분석 결과를 제시한다.

Hypertext Retrieval System Using XLinks

Eun-Jung Kim† · Jong-Min Bae††

ABSTRACT

Most of hypertext retrieval models consider documents as independent entities. They ignore relationships between documents or link semantics. In an information retrieval system for hypertext documents, retrieval effectiveness can be improved when ling information is used. Previous link-based hypertext retrieval models ignore link information while indexing. They utilize link information to re-rank the retrieval results. Therefore they are limited that only the documents in result-set utilize link information. This paper utilizes link information when indexing. We present how to use term weighting and inLinks weighting for ranking the relevant documents. Experimental results show that recall and precision evaluation according to the link semantics and the comparison with previously link_based hypertext retrieval model.

키워드 : 하이퍼텍스트(hypertext), 정보 검색 시스템(information retrieval system), XML, XLinks, 문서 색인 방법(document indexing method), 링크기반 검색 시스템(link_based retrieval system)

1. 서 론

현재 인터넷상에서 보편화된 대부분의 문서는 하이퍼텍스트 형태로 정보 검색의 대상이 되고 있다. 하이퍼텍스트 문서는 정보의 내용을 나타내는 노드와 이 정보 단위들간의 연관성을 나타내는 링크로서 이루어져 있다. 브라우징 기반의 정보 검색 시 하이퍼텍스트 문서는 비 선형적 접근을 가능하게 하며, 사용자는 링크를 통하여 서로 연결된 정보들을 검색한다. 그러나, 브라우징 기반의 검색 방법은 검색 대상인 정보 공간이 매우 큰 경우에는 노드와 링크의 수가 너무 많아 그리 효과적이지 못하여 하이퍼텍스트 시스템에 대한 질의어 기반의 검색 방법에 대한 연구가 많이 있어 왔다[2, 4]. 일반적인 질의어 기반의 하이퍼텍스트 정보 검색 시스템들은 링크 정보를 무시하고 모든 노드를 독립적인 문서로 간주하여 검색한다[3]. 따라서 하이퍼텍스트 기반의 정보 검색에 있어 링크 정보를 활용하는 연구가 많이 진행되어 왔고, 이러한 기존의 연구들의 결론은 문서내의 링크 정보를 검색에 활용하면 검색의 성능을 향상시킬 수 있다는 것이다. 링크 정보를 검색에 활용한 기존 연구들의 공통점은 문서의 색인 과정에서는 링크 정보를 무시하고 모든 노드를 독립적인 문서로 간주하여 색인 한다. 그런 다음 질의어에 대하여 문서를 검색하는 과정에서 문서의 우선 순위를 변화시키는데 링크 정보를 활용한다. 이는 링크 정보의 활용이 검색 결과 집합에 포함된 문서들에만 한정된다는 단점이 있다.

웹 상의 문서는 하이퍼텍스트 문서, 템플릿, 응용 프로그램, CGI 스크립트, JAVA 스크립트 등으로 이루어진다. 이러한 문서는 정적이고 영구적으로 존재하거나 또는 동적으로 생성된다. 즉, 웹 문서는 가상 문서의 형태를 가지고 있다. 가상 문서는 그 내용이 정적으로 존재하는 것이 아니라 내용의 일부 또는 전부가 실행 시에 구성되어지는 문서이다[5, 6]. 하이퍼텍스트 문서도 웹 문서 중 하나이므로 가상 문서의 부분 집합이다[6]. 가상 문서의 개념에서 볼 때, 하

† 준 회 원 : 부산외국어대학교 전자컴퓨터공학부 교수
†† 종신회원 : 경상대학교 컴퓨터과학과 교수
논문접수 : 2001년 2월 20일, 심사완료 : 2001년 7월 20일

이퍼텍스트 문서의 내용은 물리적으로 존재하는 정보의 내용과 사용자의 브라우징에 의해 동적으로 생성되는 내용으로 구성되어진다. 즉 하이퍼텍스트 문서에서는 일반적인 '하나의 문서'에 대한 개념이 달라졌음을 알 수 있다. 일반적인 문서에서의 '하나의 문서'는 물리적인 내용만을 말하는 것이지만 가상 문서에서의 '하나의 문서'는 논리적인 내용을 말한다. 이에 정보 검색을 위한 문서의 색인 과정에서 기존의 물리적인 내용만으로 색인을 하는 것에서 논리적인 문서의 개념에서 색인을 하는 메커니즘이 필요하다. 이에 본 논문에서는 이러한 가상 문서의 개념을 XML 링크 정보와 함께 하이퍼텍스트 정보 검색에 적용한다.

제안된 모델에서는 가상 문서의 개념을 적용하여 XML 링크 정보를 문서의 색인 과정에서 활용한다. 문서의 색인 과정에서 링크 정보를 활용함에 있어 먼저, 링크 의미 기반 문서내 용어의 가중치를 재정의 한다. 하나의 문서에서 용어의 가중치 정의에 있어서, 가상 문서의 개념에서 고려해야 할 사항인 문서내 용어의 빈도 수와 정규화 방법, 그리고 전체 문서 집합에서 용어에 대한 역 문헌 빈도 수를 재 정의한다. 다음으로 링크의 메타 데이터 기반 문서 내 inLinks의 가중치를 정의한다. 질의어에 대하여 문서의 검색 과정에서는 질의어에 대하여 문서의 내용 중심의 검색과 링크의 메타 데이터 중심의 검색을 위한 모델을 제시한다. 제안된 링크 의미 기반 검색 모델은 기존의 검색 모델에 비해 검색 성능을 향상시킬 뿐만 아니라 링크의 메타 데이터 중심의 검색을 제공하여 정보 검색의 시각을 보다 다양화할 수 있게 한다.

논문의 구성은 다음과 같다. 먼저 2장에서는 하이퍼텍스트 기반의 정보 검색에 대한 연구 중에서 링크 정보를 질의어 기반 검색에 활용하는 방법에 대한 관련 연구를 살펴보고 이러한 기존의 연구들의 공통점과 문제점을 제시하고 제안된 모델의 설계 방향을 설명한다. 3장에서는 일반 백터 검색 모델에서 사용하는 문서내 용어의 가중치를 계산하는 방법과 검색 상태 값(RSV, Retrieval Status Value)을 계산하는 방법을 설명한다. 다음으로 4장에서는 제안된 의미 기반 검색 모델에서의 색인 과정을 설명한다. 색인 과정에서 링크의 의미를 활용하여 문서 내 용어의 가중치를 재 정의하는 방법을 설명하고, 링크의 메타 데이터를 활용하여 문서 내 inLinks의 가중치를 정의하는 방법을 설명한다. 그리고 문서의 우선 순위를 위한 RSV 계산식을 제시한다. 5장에서는 제안된 링크 의미 기반 검색 모델에 대한 성능 평가를 하고 그 결과를 제시한다. 마지막으로 6장에서 결론으로 매듭을 짓는다.

2. 관련 연구 및 설계 방향

하이퍼텍스트 기반의 정보 검색에 대한 연구가 많이 진행되어 왔다[1, 4, 10, 11 ,13]. 이중에서 링크 정보를 질의어 기반 검색에 활용하는 방법에 대한 연구를 살펴보면 다음과 같다.

먼저, 질의어 기반 검색에 링크의 방향성과 직접/간접 링크 정보등을 활용하여 검색 결과 집합의 확장 및 문서의 순위를 변화시킨 검색 모델이다[13]. 이 모델에서는 세 단계로 나누어 검색을 수행한다. 첫 번째 단계에서는 일반 백터 처리 계산을 한다. 일반 백터 처리 모델을 이용하여 문서내의 각 색인어에 대한 가중치를 계산한다. 이 가중치 값을 기준으로 질의어에 대한 검색결과 집합이 만들어지고 각 문서는 질의어에 대하여 RSV(Retrieval Status Value)를 갖는다. 이렇게 얻어진 RSV 값에 따라 각 문서는 순위(rank)를 갖게 된다. 두 번째 단계는 검색결과 집합을 확장한다. 첫 번째 단계에서 나온 검색결과 집합에 대하여 링크의 방향성 정보를 이용하여 검색결과 집합에 있는 문서들 중에서 집합 내부에 있는 문서가 외부에 있는 문서를 지시하는 링크가 있으면, 해당 링크가 지시하는 외부 문서를 검색 결과 집합에 포함시킨다. 이때 외부 문서의 RSV는 연결된 내부 문서와의 유사도(similarity)를 이용하여 RSV를 구한다. 마지막으로 세 번째 단계는 링크의 효과를 적용한다. 이 단계에서는 저장된 링크 정보 즉, 직접/간접 링크 정보, 앵커가 질의어인지 비질의어 인지에 대한 정보들을 각 문서에 적용한다. 이 단계를 거치면 모든 문서의 RSV 값이 변하여 문서의 순위가 변화한다. 두 번째 모델은 링크 정보를 검색에 이용하기 위해 두 단계로 나누어 확장된 벡터 처리 기법을 사용한 방법이다[4]. 첫 번째 단계에서 일반 백터 처리 모델을 이용하여 문서의 내용에 따라서 용어의 가중치 값을 계산하고, 질의어에 대하여 문서의 RSV값을 계산하여 검색결과 집합을 생성한다. 두 번째 단계에서는 첫 번째 단계의 검색결과 집합에서 상위에 우선 순위 된 문서(예를 들면, 상위 20개의 문서)들을 기준으로 링크 정보를 이용하여 문서의 RSV값을 확장하였다. 예를 들면, 문서1에서 문서2로 링크가 설정되어 있을 때 문서1의 RSV값에 링크의 가중치를 곱한 값을 문서2의 RSV값에 더하여 문서2의 RSV를 확장하였다. 세 번째 모델은 검색 알고리즘이 내용 분석(content analysis)과 문맥 분석(context analysis)의 두 단계로 이루어져 문서의 순위를 재조정하는 모델이다[1]. 첫 번째 단계는 내용 분석 과정으로 특정 검색 시스템에 질의어를 보내어 해당 시스템으로부터 검색된 질의어에 관련된 문서들 중에서 상위에 우선 순위 된 문서들(예를 들면, 상위 200개의 문서들)을 선택하여 검색 결과 집합을 생성한다. 두 번째 단계에서는 검색 결과 집합의 문서들을 재 랭킹하기 위하여 링크 정보를 활용한 문맥 분석 과정으로서 문서 내 inLinks(외부에서 내부로 들어오는 링크)의 개수와 outLinks(내부에서 외부로 나가는 링크)의 개수를 이용하여 문서의 인기도(popularity, 외부 문서들로부

터 링크를 받고 있는 정도)를 조사 위해 각각 독립적으로 수행되는 7개의 변화 알고리즘을 개발하였다. 두 번째 단계에서 나온 각 문서의 인기도를 바탕으로 문서들의 우선 순위를 재조정하였다. 이러한 기존 연구들의 공통점은 먼저, 문서의 색인 과정에서는 문서 내 링크 정보를 무시하고 모든 문서를 독립적인 문서로 간주하여 색인 한다. 색인 결과 문서내 용어의 가중치 값이 계산되고 이 가중치 값을 기준으로 사용자의 질의어에 대한 검색 결과 집합이 만들어진다. 다음으로 검색 결과 집합에 포함된 문서들을 기준으로 문서의 우선 순위를 조정하는데 링크 정보를 활용한다. 즉, 색인 과정이 끝나고 질의어에 대하여 문서를 검색하는 과정에서 문서의 재 랭킹을 위해 링크 정보를 이용한다. 이는 링크 정보가 검색 결과 집합에 포함된 문서들에만 한정되어 활용된다는 단점이 있다. 따라서 전체 문서 집합에 있는 모든 문서들을 대상으로 링크 정보를 활용할 수 있는 메커니즘이 필요하다.

이전의 연구에서 이미 XML 링크 정보를 이용한 정보 검색 색인 기법과 링크의 메타 데이터를 이용한 검색 시스템을 설계한 바 있다[12, 14, 15]. 이를 바탕으로 제안된 모델에서는 XML 링크 정보를 문서의 색인 과정에서 활용하여 문서내 용어의 가중치와 문서 내 inLinks의 가중치를 정의한다. 먼저, 문서내 용어의 가중치를 계산함에 있어 가상 문서의 개념을 이용하여 계산한다. 즉 하나의 문서를 색인할 때, 지역 문서뿐만 아니라 문서 내 링크가 가리키고 있는 원격 문서도 지역 문서의 일부분으로 간주하고 지역 문서의 색인에 포함시킨다. 이를 위해 링크의 유형과 행동에 관련된 속성들을 기준으로 링크의 의미를 분류하여 고유한 식별자를 부여하고, 각 식별자가 갖는 행동의 의미를 파악하여 링크의 가중치를 부여하였다. 정의한 링크의 가중치 값을 이용하여 문서내 용어의 가중치를 계산한다. 이때 고려해야 할 사항으로 문서 내 용어의 빈도 수, 정규화 방법, 용어에 대한 역 문헌 빈도 수를 들 수 있다.

다음으로 문서 내 inLinks의 가중치를 계산하기 위하여 XML 링크의 메타 데이터를 이용한다. XML 링크는 ROLE 속성을 이용하여 자신이 지시하는 원격 문서의 역할을 설명한다. 즉, ROLE 속성 값은 원격 문서의 메타 데이터로서의 작용을 한다. 이러한 ROLE 속성 값을 이용하여 문서의 색인 과정에서 하나의 문서 내로 들어오는 inLinks의 개수를 주제별로 검색하여 계산한다. inLinks의 개수를 기반으로 질의어에 대하여 문서 내 inLinks의 가중치를 계산한다.

문서의 검색 과정에서는 내용 중심의 검색과 링크의 메타 데이터 중심의 검색을 위한 방법을 제시한다. 내용 중심의 검색은 질의어와 가장 가까운 내용의 문서를 찾는 검색으로서 문서 내 용어의 가중치와 문서 내 inLinks의 가중치를 이용하여 정의하고, 링크의 메타 데이터 중심의 검색은 질의어와 관련된 내용으로 가장 많은 링크를 설정 받은 문서를 찾는 검색으로 문서 내 inLinks의 가중치를 이용하여 정의한다.

3. 일반 벡터 검색 모델

문서나 하이퍼텍스트 노드 안에 있는 개념들이나 또는 각 단일 용어들이 갖는 상대적인 중요성을 측정하기 위해 일반 벡터 공간 모델에서는 다음의 작업을 수행한다. 문서 i에서 용어 j에 대한 가중치를 계산하는 방법은 식 (1)과 같다.

$$w_{ij} = tf_{ij} \cdot idf_j = tf_{ij} \cdot \log\left[\frac{n}{df_j}\right] \quad (1)$$

식 (1)에서, w_{ij}는 문서 i에서 용어 j의 가중치이며 , tf_{ij}는 문서 i에서 용어 j의 빈도 수, idf_j는 용어 j에 대한 역 문헌 빈도 수이다. n은 전체 문서 수, df_j는 용어 j를 가지는 문서 수이다. 식 (1)에서 문서의 길이를 고려한 문서 i에서 용어 j의 정규화 된 가중치는 다음의 식 (2)와 같다.

$$w_{ij} = ntf_{ij} \cdot nidf_j = \frac{tf_{ij}}{\max tf_i} \cdot \frac{\log\left[\frac{n}{df_j}\right]}{\log(n)} \quad (2)$$

식 (2)에서 ntf_{ij}는 정규화 된 총 빈도 수, tf_{ij}는 문서 i에서 용어 j에 대한 총 빈도 수, max tf_i는 문서 i에서 가장 큰 용어의 빈도 수, $nidf_j$는 정규화 된 idf을 나타내기 위해 전체 문서 집합의 길이에 대한 대수로서 나누었다. 식 (2)의 결과로 벡터 모델에 의한 질의어에 대한 검색 결과 집합이 만들어진다. 식 (2)의 결과로 나온 검색 결과 집합에서, 질의어 Q에 대하여 문서의 순위를 계산하기 위하여 각 문서의 RSV를 계산한다. 각 문서 D_i의 RSV(Retrieval Status Value)의 계산은 식 (3)과 같다.

$$RSV(D_i,Q) = \sum_{j=1}^{q} w_{ij} \cdot w_{qj} \text{ for } i = 1,2,\cdots,n \quad (3)$$

각 문서 D_i는 질의어 Q에 대하여, 질의어 Q에 있는 모든 용어 j의 문서 가중치 w_{ij}와 질의 가중치 w_{qj}의 곱에 대한 누적 합산의 결과로 RSV를 갖는다. w_{ij}는 식 (1)에서 정의한 내용이고, w_{qj}는 질의어 q에서 용어 j가 갖는 가중치이다. 이렇게 얻어진 RSV값에 따라 각 문서는 질의어 Q에 대해 순위를 갖게 된다.

4. 링크 기반 색인을 위한 가중치 정의

4.1 문서내 용어의 가중치 재정의

제안된 모델에서는 문서내 용어의 가중치를 정의함에 있어 가상 문서의 개념을 적용하여 링크 정보를 문서의 색인 과정에서 이용한다. 따라서 하나의 문서를 색인 할 때, 지역 문서뿐 아니라 링크가 지시하는 원격 문서도 지역 문서

의 일부로 간주하여 색인 한다. 문서내의 링크가 지시하는 원격 문서의 내용을 가상적으로 지역 문서의 일부분으로 간주하지만, 물리적으로는 독립적으로 존재하는 내용이므로 지역 문서에 포함된 원격 문서의 가중치를 위하여 두 문서 사이에 정의된 링크의 속성을 이용한다.

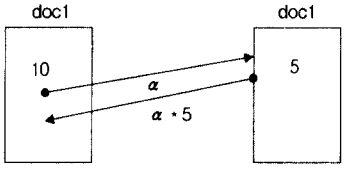
링크는 그 속성에 따라 관련성의 정도를 다르게 정의할 수 있다[7,8]. 따라서 링크는 정의된 속성별로 다양한 종류가 있다. 제안된 모델에서는 링크의 종류를 분류하기 위해 링크의 속성 중에서 유형을 정의하는 type 속성과 링크의 행동을 정의하는 show, actuate 속성을 이용한다. 각 속성 값에 따라 링크를 분류하여 식별자(ID)를 <표 1>과 같이 정의한다. <표 1>에서 보이듯이, 링크의 유형 중에서 단순 링크와 인라인 확장 링크만을 고려하였으며, 각 링크의 행동에 따라 고유한 식별자를 부여한다. 그리고 각 식별자의 행동이 갖는 의미에 따라 지역 문서와 원격 문서사이의 관련성의 정도를 파악하여 링크의 가중치를 부여한다.

〈표 1〉 링크의 식별자 테이블

링크의 속성			식별자	가중치	링크의 행동
TYPE	ACTUATTE	SHOW			
Simple/ inline_ Extended	OnLoad	embed	1	α	원격 문서가 자동적으로 지역 문서에 삽입됨
	OnLoad	replaced	2	β	지역 문서가 자동적으로 원격 문서로 대체됨
	OnLoad	new	3	γ	원격 문서가 자동적으로 새로운 윈도우에 나타남
	OnRequest	embed	4	δ	사용자의 선택에 의해 삽입됨
	OnRequest	new	5	ε	사용자의 선택에 의해 나타남
	OnRequest	replaced	6	θ	사용자의 선택에 의해 대체됨

4.1.1 문서내 용어의 빈도 수

문서내 용어의 빈도 수를 계산할 때 지역 문서뿐만 아니라, 지역 문서내의 링크가 지시하는 원격 문서에 있는 용어도 지역 문서의 용어 빈도 수 계산에 포함한다. 이때 지역 문서에 있는 용어의 빈도 수를 1로 부여할 때 원격 문서에 있는 용어의 빈도 수는 두 문서간의 링크의 식별자에 따라 서로 다르게 부여한다. 즉, 링크의 식별자가 갖는 가중치 값을 원격 문서에 있는 용어의 빈도 수 계산에 이용한다. 예를 들어 (그림 1)에서 doc1에서 특정 용어가 10번, doc1에서 지시하는 doc2에 해당 용어가 5번 존재한다. 이때 링크의 가중치가 α인 경우, doc2에 있는 하나의 용어는 α만큼 계산되어 doc1의 빈도 수에 계산된다.



doc1의 용어의 빈도수 = $10 + \alpha * 5$

(그림 1) 용어의 빈도수 계산

따라서, 제안된 모델에서 용어의 빈도 수 계산 공식은 다음의 식 (4)와 같다.

$$tf'_{ij} = \text{local } tf_{ij} + \text{remote } tf_{ij}$$

$$\text{where remote } tf_{ij} = \alpha \cdot \sum_{\alpha=1}^{r} tf_{\alpha j} + \beta \cdot \sum_{\beta=1}^{s} tf_{\beta j} + \gamma \cdot \sum_{\gamma=1}^{t} tf_{\gamma j} + \delta \cdot \sum_{\delta=1}^{w} tf_{\delta j} + \varepsilon \cdot \sum_{\varepsilon=1}^{e} tf_{\varepsilon j} + \theta \cdot \sum_{\theta=1}^{u} tf_{\theta j} \quad (4)$$

식 (4)에서 local tf_{ij}는 지역 문서 i에서 용어 j의 빈도 수이고 remote tf_{ij} 는 지역 문서 i의 링크가 지시하는 모든 원격 문서에서의 용어 j의 빈도 수이다. α, β, γ, δ, ε, θ는 링크의 가중치이며, r, s, t, w, e, u는 각 가중치에 해당하는 링크의 개수이다. $\alpha \cdot \sum_{\alpha=1}^{r} tf_{\alpha j}$는 지역 문서에 포함된 링크들 중에서 링크의 가중치가 α인 모든 원격 문서에서의 용어 j의 빈도 수이며, 그 외의 경우도 각각의 링크의 가중치가 갖는 모든 원격 문서에서의 용어 j의 빈도 수이다. 여기서 $tf_{\alpha j}$는 계속하여 확장되어 계산되어진다. 자세한 내용은 4.1.2에서 설명한다.

4.1.2 다단계 링크의 제어 속성 정의

링크가 지시하는 원격 문서 안에는 계속해서 링크가 포함되어 있을 수 있다. XML 링크의 속성에는 링크 된 원격 문서 안에 있는 링크에 대해서 지역 문서에서 제어할 수 있는 방법은 제공되지 않는다. 예를 들어, 지역 문서에 있는 링크들 중에서 속성이 show가 'embed'이고 actuate가 'onLoad'인 링크가 있다고 가정하자. 이 링크는 지역 문서가 활성화될 때 링크 된 원격 문서의 내용이 자동적으로 지역 문서에 삽입되게 된다. 이때 삽입된 원격 문서 안에 다시 똑같은 링크가 존재한다면 해당 링크가 지시하는 원격_원격 문서의 내용도 자동적으로 지역 문서 안에 삽입된다. 이런 경우가 다단계 링크의 경우이다. 지역 문서 작성자가 원격 문서 안에 있는 해당 링크가 지시하는 원격_원격 문서는 지역 문서가 활성화될 때 자동적으로 삽입되기를 원하지 않더라도 XLink에는 원격 문서 안의 링크에 대한 제어 속성이 없기 때문에 사용자가 원격 문서 안의 링크에 대해서 활성화 여부를 제어할 수 없다. 제안된 모델에서는 원격 문서 안의 어떤 링크에 대해서 지역 문서에서 해당 링크를 제어할 수 있는 새로운 속성을 정의한다. 정의하는 속성은 다음과 같다.

(1) remote_link : 원격 문서 안의 링크를 그대로 포함시킬지의 여부를 정의하는 속성으로서 가질 수 있는 값은 yes와 no이다. yes값은 링크 된 원격 문서 안의

모든 링크를 그대로 수용한다. remote_ link에 대한 정의가 생략되었다면 디폴트값이 yes이다. no값은 링크 된 원격 문서 안의 링크들을 그대로 수용하지 않고, 수용할 링크만 insertion_link 속성에서 정의한다.

(2) insertion_link : remote_link의 값이 no일 경우, 링크된 원격 문서 안에 있는 링크를 모두 수용하지 않는다. 이때 사용자가 수용하고자 하는 링크만 이 속성에서 정의한다. 이 속성이 가질 수 있는 값은 링크에 대한 ID이다. ID 값은 본 논문에서 정의하는 링크 식별자이다. 이 속성에서 수용하고자 하는 해당 식별자를 선택하여 열거한다. 이 속성에서 열거된 링크들만이 원격 문서 안에서 원래의 속성대로 활성화된다.

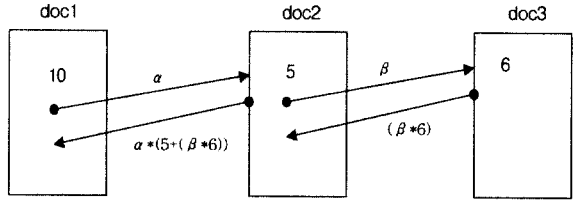
정의한 remote_link와 insertion_link의 속성 값에 따라 원격 문서에서의 링크가 지시하는 원격_원격 문서에서도 계속해서 용어의 빈도를 계산할 것인지를 결정한다. 즉 지역 문서에서의 링크가 갖는 속성 값에 따라 원격 문서 안의 링크는 가중치를 가지며, 가중치를 가지는 링크가 지시하는 원격_원격 문서는 용어의 빈도를 계산하는데 포함된다. 원격 문서 안의 링크에 대한 가중치 부여는 <표 2>와 같다.

〈표 2〉 원격 문서안의 링크 가중치

ID	지역문서안의 링크 속성값	원격 문서안의 링크 가중치
1	remote_link = 'yes'	α
	remote_link = 'no' and insertion_link = '1'	α
2	remote_link = 'yes'	β
	remote_link = 'no' and insertion_link = '2'	β
3	remote_link = 'yes'	γ
	remote_link = 'no' and insertion_link = '3'	γ
4	remote_link = 'yes'	δ
	remote_link = 'no' and insertion_link = '4'	δ
5	remote_link = 'yes'	ε
	remote_link = 'no' and insertion_link = '5'	ε
6	두 개의 속성값에 무관	0

원격 문서 안에 있는 링크는 <표 2>를 기준으로 가중치 값을 가지며, 링크가 가중치 값을 가지면 원격_원격 문서에서도 용어의 빈도를 계산한다. 그러나 원격 문서에 있는 ID가 6인 링크는 지역 문서의 링크의 속성 값에 상관없이 더 이상 가중치 값을 가지지 않는다. 즉 원격 문서에 있는 ID가 6인 링크가 지시하는 원격_원격 문서는 용어 빈도 수 계산에 포함되지 않는다.

정의한 속성과 링크의 가중치 값에 따라 문서의 색인 시, 다단계 링크에 대한 원격 문서, 원격_원격 문서의 용어 빈도 수는 해당 링크의 가중치 값을 기준으로 지역 문서의 용어 빈도 수에 계산된다. 지역 문서에 있는 특정 용어에 대한 빈도 수를 계산함에 있어서 지역 문서가 지시하는 원격 문서에서의 용어의 빈도 수는 지역 문서의 링크 가중치 값으로 계산되고, 원격_원격 문서에서의 용어의 빈도 수는 지역 문서의 링크 가중치 값과 원격 문서에서의 링크 가중치 값을 곱하여 계산된다. 예를 들어, (그림 2)에서 doc1의 특정 용어에 대한 빈도 수를 계산함에 있어서, doc2에서의 용어의 빈도 수는 지역 문서에 있는 링크의 가중치 α로서 계산되며, doc3에서의 용어의 가중치는 지역 문서에 있는 링크의 가중치 α와 원격 문서에 있는 링크의 가중치 β로서 계산된다. 그 결과 doc2와 doc3에서의 용어의 빈도수가 doc1의 용어의 빈도 수에 계산된다.



doc1의 용어의 빈도수 = $10+\alpha*(5+(\beta*6)$

(그림 2) 다단계 링크의 용어 빈도 수

이에 제안된 모델에서는 다단계 링크에 의한 원격_원격 문서에서의 용어의 빈도 수 계산을 식 (5)와 같이 정의한다.

$$tf_{aj} = \text{remote } tf_{aj} + \text{remote\_remote } tf_{aj}$$

$$\text{where remote\_remote } tf_{aj} = \alpha \cdot \sum_{\alpha=1}^{r} tf'_{aj} + \beta \cdot \sum_{\beta=1}^{s} tf'_{\beta j} + \gamma \cdot \sum_{\gamma=1}^{t} tf'_{\gamma j} + \delta \cdot \sum_{\delta=1}^{w} tf'_{\delta j} + \varepsilon \cdot \sum_{\varepsilon=1}^{e} tf'_{\varepsilon j} \quad (5)$$

식 (5)에서 tf_{aj}는 식 (4)에서 정의한 지역문서의 링크 중에서 가중치가 α인 링크가 지시하는 원격 문서에서의 용어의 총 빈도 수이다. remote tf_{aj}는 원격 문서 내에서의 용어의 빈도 수이고, remote_remote tf_{aj}는 원격 문서의 링크가 지시하는 모든 원격_원격 문서 내에서의 용어 j의 빈도 수이다. α, β, γ, δ, ε는 원격 문서 내 링크의 가중치이며, x, y, z, q, w는 각 가중치에 해당하는 링크의 개수이다. $\alpha \cdot \sum_{\alpha=1}^{r} tf'_{aj}$는 원격 문서에 포함된 링크들 중에서 링크의 가중치가 α인 모든 원격_원격 문서에서의 용어 j의 빈도 수이며, 다른 경우도 각각의 링크의 가중치가 갖는 모든 원격_원격 문서에서의 용어 j의 빈도 수이다.

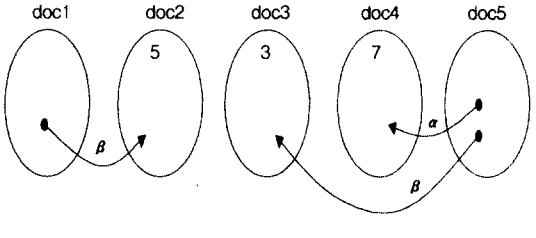
4.1.3 용어에 대한 역 문헌 빈도 수

용어에 대한 역 문헌 빈도 수(IDF, Inverse Document Frequency)는 전체 문서 집합에서 용어가 갖는 중요도를 평가하는데 사용된다. 일반적으로 전체 문서 집합에서 적은 수의 문서에서 쓰이는 용어는 많은 수의 문서에서 쓰이는 용어에 비해 그 중요도를 더 높이 평가한다. 즉 자주 쓰이지

않는 용어가 자주 쓰이는 용어에 비해 그 중요도가 더 높다고 할 수 있다. 일반 벡터 모델에서는 식 (1)에서 정의하듯이 이러한 용어의 역 문헌 빈도 수를 전체 문서 집합에서 해당 용어를 포함한 문서의 개수를 이용하여 정의한다.

가상 문서에서는 용어가 나타나는 문서를 지역 문서에 용어가 나타나는 문서뿐만 아니라 지역 문서에는 용어가 없어도 원격 문서에 용어가 나타나는 문서와 원격_원격 문서에 용어가 나타나는 문서의 경우에도 모두 용어가 나타나는 문서로 간주한다. 따라서 가상 문서에서는 용어가 나타나는 문서의 수(df)가 많이 늘어나게 된다. 이러한 변화가 갖는 의미는 다음과 같다. 첫째, 전체 문서 집합에서 용어에 대한 IDF를 변화시킴으로 인해서 전체 문서 집합에서 해당 용어에 대한 중요도를 변화시킨다. 둘째, 전체 문서 집합에서 두 용어사이의 IDF 값의 크기 순서를 변화시킴으로 인해서 전체 문서 집합에서 두 용어간의 중요도의 순위를 변화시킨다. 셋째, IDF의 변화에 따라서 하나의 문서에서 용어의 가중치를 변화시킨다. 넷째, 이러한 용어의 가중치 값의 변화로 인해서 하나의 문서에서 두 용어사이의 가중치 값의 크기 순서가 변화시키고, 이는 하나의 문서가 갖는 역할을 변화시킨다. 따라서 이 모든 변화로 인해서 질의어에 대하여 문서의 순위가 변화될 수 있다.

이에 제안된 모델에서는 df를 재 정의한다. 즉 지역 문서에는 용어가 없고 원격 문서에만 용어가 있는 경우의 문서의 빈도를 계산함에 있어 링크의 가중치를 이용한다. 지역 문서에 용어를 포함하는 문서를 빈도 1로 계산할 때, 지역 문서에는 용어가 없고 원격 문서 또는 원격_원격 문서에 용어가 포함된 문서의 빈도는 링크의 가중치를 이용하여 계산한다. 예를 들어 (그림 3)에서 문서 doc2, doc3, doc4는 지역 문서에 용어 j를 포함하고 있다. doc1은 지역 문서에는 용어 j를 포함하고 있지 않지만 doc2로 링크를 설정하고 있기 때문에 용어 j를 포함하는 것으로 간주한다. doc5의 경우에도 지역 문서에는 용어 j가 없어도 doc3, doc4로 링크를 설정하고 있기 때문에 용어 j를 포함하는 것으로 간주한다. 따라서 전체 문서 집합에서 용어가 나타나는 문서의 개수를 계산함에 있어 용어 j를 포함한 문서의 개수가 3과 doc1의 가중치 값 β 그리고 doc5의 가중치 값중 큰 것을 기준으로 α로서 계산된다.



df_j = 3(doc2, doc3, doc4) + β(doc1의 빈도) + α(doc5의 빈도)

(그림 3) 용어 j를 포함하는 문서의 빈도

따라서 제안된 모델에서는 하나의 용어에 대한 역 문헌 빈도 수를 식 (6)과 같이 확장하여 정의한다.

$$idf_j = \log\left[\frac{n}{dfj}\right]$$

$$\text{where } df_j = df'_j + \sum_{d=0}^{k} (\text{MaxLinkWeight}) \qquad (6)$$

식 (6)에서 df'_j는 지역 문서에 용어 j를 포함하는 문서의 개수이고, n은 전체 문서의 개수, 그리고 d는 지역 문서에는 용어 j가 없어도 원격 문서에 용어 j를 포함하는 문서이며, k는 d 경우의 문서의 개수이다. 따라서 $\sum_{d=0}^{n}$(MaxLinkWeight)는 지역 문서에는 용어 j를 포함하지 않고 대신 원격 문서에 용어 j를 포함하는 문서들의 개수를 링크의 가중치를 이용하여 계산한 개수이다. MaxLinkWeight는 하나의 문서에서 용어가 있는 원격 문서를 지시하는 링크가 여러 개 있을 때, 그 중에서 가장 큰 링크의 가중치 값을 의미한다.

4.1.4 정규화

실제 물리적인 하나의 문서를 가상 문서의 개념에서 보면, 문서의 길이가 많이 늘어나게 된다. 하나의 문서가 길이가 늘어나는 것에 대해서 함께 늘어나는 최대 용어 빈도 수를 이용하여 문서의 길이를 제어할 수 있다. 따라서 이러한 최대 용어 빈도 수를 이용하여 늘어난 문서의 길이에서 용어 빈도 수를 일반 벡터 모델의 식 (7)과 같이 정규화하여 정의한다.

$$tf_{ij} = \frac{tf_{ij}}{MAXtf_i} \qquad (7)$$

식 (7)에서 $MAXtf_i$는 지역 문서 i와 문서 i에서 지시하는 모든 원격, 원격_원격 문서에서의 가장 큰 용어 빈도 수이다. 제안된 모델에서 역 문헌 빈도 수에 대한 정규화도 일반 벡터 모델에서의 정규화 방법을 사용하여 정의한다.

4.2 문서 내 inLinks 가중치 정의

제안된 모델에서는 문서의 색인 과정에서 문서 내 inLinks의 가중치를 새롭게 정의한다. XML 링크에는 링크에 참여하는 자원의 의미(semantic)와 관련된 속성, ROLE 이 있다. 이 속성은 일반적으로 링크 된 원격 문서의 역할을 설명하기 위하여 사용된다. 따라서 이러한 role 속성 값이 가지는 의미는 특정 단어를 메타 데이터로 가지는 링크가 지시하는 원격 문서는 메타 데이터와 관련 있는 내용임을 의미한다. 이 의미를 검색에 활용하기 위하여 제안된 모델에서는 링크의 ROLE 속성을 이용하여 하나의 문서에서 특정 메타 데이터를 갖는 inLinks가 문서에서 갖는 가중치를 정의한다. 이를 위해 모든 문서의 색인 과정에서 outLinks

(문서내에서 다른 문서를 지시하는 링크)에 대한 정보를 메타 데이터와 함께 별도의 영역에 저장한다. 그 결과 각 문서는 inLinks에 대한 정보를 가지게 되고, 이러한 정보를 이용하여 하나의 문서에서 서로 다른 메타 데이터를 가진 링크들의 가중치를 정의한다.

문서 내 특정 용어에 대한 inLinks의 가중치를 정의함에 있어 문서 내 incoming link의 개수를 이용한다. 특정 메타 데이터를 가진 incoming link의 개수가 많다는 것은, 외부의 다른 문서에서 해당 문서를 참조할 때 같은 내용의 메타 데이터로서 많은 참조를 하고 있다는 의미이다. 따라서 해당 문서는 그 메타 데이터와 아주 가까운 내용의 문서로 생각할 수 있다. 즉, 특정 메타 데이터를 가진 incoming link의 개수가 많다는 것은 해당 문서가 해당 메타 데이터에 가까운 내용임을 의미한다. 이 경우, 하나의 문서 안에서 특정 용어에 대한 incoming link의 개수가 갖는 의미는 같은 문서 내 다른 용어에 대한 incoming link의 개수에 영향을 받는다. 따라서 하나의 문서에서 특정 메타 데이터에 대한 inLinks의 가중치는 다른 메타 데이터의 incoming link의 개수를 고려하여 정의해야 한다.

이를 위해 제안된 모델에서는 문서 내 특정 용어에 대한 inLinks의 가중치를 정의함에 있어, 식 (8)에서와 같이 정의한다. 문서 내 특정 용어에 대한 inLinks의 가중치를 해당 문서에서 다른 용어를 가진 가장 많은 incoming link의 개수를 이용하여 상대적인 값으로 정의한다.

$$inLinkWeight'_{im} = \frac{inLinkCount_{im}}{\max\ inLinkCount_i} \qquad (8)$$

4.3 문서의 우선 순위를 위한 RSV 계산 방법

사용자의 질의어에 대하여 문서의 내용을 중심으로 한 검색과 링크의 메타 데이터를 중심으로 한 검색을 제공한다.

4.3.1 문서 내용 중심의 검색

질의어에 대하여 문서 내용 중심의 검색을 위해서 제안된 모델에서는 두 가지의 RSV 계산식을 제안한다. 첫째, 일반 벡터 모델에서와 같은 일반적인 RSV 계산 방법이다. 질의어 Q에 대하여 문서 D_i의 일반적인 RSV는 식 (3)에서와 같이 문서에서의 용어의 가중치 w_{ij} 와 질의어 q에서 용어 j의 가중치 w_{qj}를 곱하여 누적 합산의 결과로 계산한다. 이렇게 계산된 각 문서는 RSV 값에 따라 문서는 우선 순위를 가지게 된다. 둘째, 문서 내 inLinks 의 가중치를 이용한 확장된 RSV 계산 방법이다. 질의어에 대하여 문서의 우선 순위를 정의할 때, 문서 내 용어의 가중치와 문서 내 inLinks의 가중치를 같이 이용하여 정의한다. 이는 문서 내용이 질의어와 얼마나 가까운 내용이지에 대한 고려와 함께 해당 문서가 질의어에 관련된 내용으로 얼마나 많은 링크를 설정 받았는지를 같이 고려할 수 있다. 즉, 문서 내용이 질의어와 아주 가까운 내용 중에서 다른 사람의 관심을 더 많이 받은 문서에 보다 높은 우선 순위를 부여할 수 있다. 확장된 RSV 계산 방법은 식 (9)와 같이 정의한다. 식 (9)에서 질의어 Q에 대하여 문서 D_i의 확장된 RSV는 문서에서의 용어의 가중치 w_{ij}와 질의어 q에서 용어 j의 가중치 w_{qj}를 곱한 값에 문서 내 inLinks의 가중치를 더하여 누적 합산의 결과로 계산된다.

$$RSV(D_i,Q) = \sum_{j=1}^{q} (w_{ij} \cdot w_{qj} + inLinkWeight_{ij}) \qquad (9)$$

식 (9)에서 사용자의 질의어에 대하여 문서의 RSV를 계산함에 있어, 문서내 용어의 가중치를 이용한 RSV 값에 해당 질의어를 메타 데이터로 가지는 incoming link가 있다면, inLinks의 가중치 값을 더하여 계산한다. 이때 해당 용어를 메타 데이터로 갖는 inLinks의 가중치가 없을 경우에는 식 (3)의 정의에 따라 문서내 용어의 가중치 값만을 이용하여 RSV를 구한다. 이렇게 얻어진 RSV 값에 따라 질의어에 대하여 문서의 우선 순위를 부여한다.

4.3.2 링크의 메타 데이터 중심의 검색

질의어에 대하여 링크의 메타 데이터 중심의 검색은 질의에 포함된 용어를 메타 데이터로 가지는 링크를 가장 많이 설정 받은 문서 위주로 순위를 부여한다. 즉 다른 사람들로부터 해당 질의어로서 가장 많은 관심을 받는 문서를 찾는 검색이다. 이는 문서 내 용어의 가중치보다는 문서 내 inLinks 의 가중치에 더 중점을 둔다. 이를 위해 질의어에 대하여 문서의 LinkRSV를 식 (10)과 같이 정의한다.

$$LinkRSV(D_i,Q) = \sum_{j=1}^{q} inLinkWeight_{ij} \qquad (10)$$

식 (10)에서 질의어 Q에 대하여 문서 D_i의 LinkRSV는 질의어 Q에 포함된 용어에 대하여 문서에서의 inLinks의 가중치를 누적 합산하여 그 결과 값으로 계산된다. 이렇게 얻어진 LinkRSV값에 따라 문서의 우선 순위를 부여한다.

5. 실험 및 분석

여기서는 제안된 XML 링크의 의미를 기반으로 한 검색 모델의 성능을 평가하기 위하여 실험 결과를 제시하고 분석한다. 정확한 분석을 위해서는 XML 문서 집합을 대상으로 해야 하지만, 현재 웹 상의 대부분의 문서가 HTML 문서이고 XLink를 사용한 적당한 XML 문서 집합의 부족으로 인해 제안된 모델의 실험을 위해 HTML 문서 집합을 이용하였다. 실험에 이용할 HTML 문서 집합은 검색기 '포탈 알타비스타'와 '야후코리아'를 이용하여 질의어 'xml', 'html', 'sgml'의 검색 결과 중에서 각 상위에 우선 순위 된

문서들을 대상으로 랜덤 하게 선택하여 전체 문서 집합을 구성하였다. 이렇게 구성된 전체 문서 집합을 대상으로 제안된 모델에 대한 성능을 평가하기 위하여 몇 가지 실험을 위한 가정을 하고, 여러 단계에서 실험 및 분석을 하였다.

5.1 실험을 위한 가정

HTML 링크의 유형이 XML 링크의 유형과 다르기 때문에 제안된 모델의 평가를 위하여, 전체 문서 집합의 문서들을 분석하여 제안된 모델에서의 링크의 식별자를 가정하여 부여하였다. 예를 들어, 스크립트 파일을 이용하여 자동으로 윈도우를 디스플레이 한 경우의 링크는 ID(3)으로 가정하고, META 태그에 속성을 지정하여 자동으로 다른 페이지를 디스플레이 해 주는 링크는 ID(2)로 가정한다.

링크의 가중치 값은 해당 링크의 의미를 판단하여 두 문서간의 관련성의 정도를 파악하여 <표 3>과 같이 정의한다. 예를 들면, 지역 문서에 나오는 용어의 빈도를 1로 부여할 때, ID(1)은 지역 문서가 활성화되면 원격 문서가 자동으로 지역 문서에 삽입되어 나타나므로 원격 문서에 나타나는 용어가 지역 문서와 관련성의 정도가 아주 높다고 판단하여 1을 부여하고, ID(6)의 경우에는 두 문서간에 독립적인 성격이 강한 경우에 많이 사용하므로 관련성의 정도가 다른 링크에 비해 상대적으로 낮다고 판단하여 0.2를 부여한다.

다단계 링크들에 대한 색인 여부를 판단하기 위해, 제안된 모델에서 제시하는 remote_link와 insertion_ link의 정의를 이용하여 <표 3>과 같이 가정하여 정의한다. 예를 들면, ID(1)과 같이 자동 삽입되는 링크는 자신이 지시하는 원격 문서 내에 있는 링크 중에서 수동으로 활성화되는 링크만 포함한다. 즉, 자신이 자동으로 삽입되는 링크이므로 자신이 활성화되면서 원격 문서에 있는 자동으로 활성화되는 링크는 활성화시키지 않는다.

〈표 3〉 링크 가중치 값과 속성값

ID	링크의 가중치	remote_link	insertion_link
1	α (= 1)	no	4, 5, 6
2	β (= 1)	yes	.
3	γ (= 0.8)	yes	.
4	δ (= 0.7)	no	4, 5, 6
5	ε (= 0.5)	yes	.
6	θ (= 0.2)	yes	.

링크의 메타 데이터는 HTML 문서 집합에서 링크의 앵커로서 가정하였으며, 링크의 앵커가 적당하지 않을 시에는 임의대로 적당한 메타 데이터를 가정하였다.

5.2 링크 유형별 검색 정확도에 대한 분석

여기서는 제안된 모델이 링크의 유형에 미치는 영향을 분석하기 위하여 링크의 유형에 따른 문서내 용어의 가중치 값의 변화를 분석하고, 이 변화에 따른 검색 정확도를 실험하였다. 실험에 사용한 문서내 용어의 가중치를 계산하는 모델은 다음과 같다.

① $w_{ij} = tf_{ij} \cdot idf_j$: 일반 벡터 처리 모델에서 정규화를 하지 않은 모델이다.

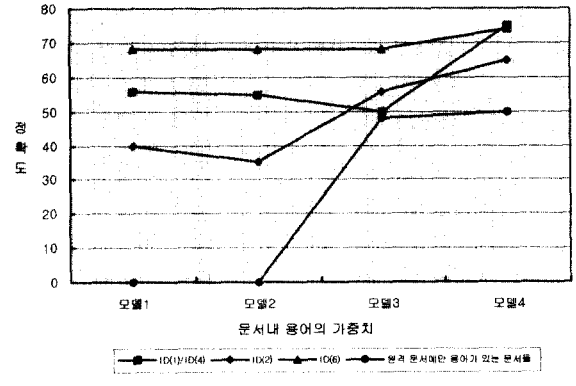
② $w_{ij} = ntf_{ij} \cdot nidf_j$: ①을 정규화 한 모델이다.

③ $w_{ij} = tf'_{ij} \cdot idf'_j$: ①에서 가상 문서의 개념을 적용하여 문서의 가중치를 계산한 모델이다. idf'_j는 재정의하지 않은 모델이다.

④ $w_{ij} = ntf'_{ij} \cdot nidf'_j$: ③을 정규화 하여 계산한 모델이다. ntf'_{ij} 는 문서 내 최대 용어 빈도수로 나누어 계산하고, $nidf'_j$는 지역 문서에 용어가 없는 문서의 빈도를 링크의 가중치를 이용하여 재 정의한 모델이다.

링크의 유형에 따라서 제시한 네 가지 모델에서의 가중치 변화와 각 가중치 값을 기반으로 질의어에 대하여 문서의 RSV를 계산하여, 이 값을 토대로 검색 정확도를 분석하였다. 질의어에 대한 각 문서의 RSV 계산은 식 (3)을 이용하였다. 식 (3)에서 문서내 용어의 가중치 w_{ij}는 제시한 네 가지 모델을 이용하였으며, 질의어에서의 용어의 가중치 w_{qj}는 binary(0 or 1)로 가정하였다.

실험을 위하여 전체 문서 집합에서 각 링크 식별자별로 해당 링크를 많이 포함하는 문서 위주로 전체 문서 집합을 다시 구성하여 각각에서의 검색 조회율과 정확도를 분석하였다. 정확도에 대한 분석 결과는 (그림 4)와 같다. (그림 4)에서 모든 링크 식별자가 전반적으로 가상 문서의 개념을 사용한 모델 3과 모델 4에서 보다 높은 성능을 나타냄을 확인할 수 있다. 따라서 링크마다 서로 의미가 다르며, 그 의미를 적절하게 반영하여 가상 문서의 개념을 적용하면 검색에 있어 효율을 증가시킬 수 있다.



(그림 4) 링크 유형별 검색 정확도의 성능

다음으로 위의 모든 종류의 링크를 포함하고 있는 전체

문서 집합에서의 검색 조회율과 정확도에 대한 분석 결과는 <표 4>와 같다.

〈표 4〉 전체 문서 집합에 대한 성능

모델	Recall(%)	Precision(%)
② $w_{ij} = ntf_{ij} \cdot nidf_j$	70.0	62.0
④ $w_{ij} = ntf'_{ij} \cdot nidf'_j$	***90.0***	***73.8***

5.3 링크의 가중치 값에 따른 검색 정확도 분석

5.2의 분석에서 링크의 가중치 값은 해당 링크가 갖는 의미를 파악하여 링크마다 다른 가중치 값을 부여하여 실험하였다. 여기서는 제안된 모델에서 링크의 가중치 값이 문서 내 용어의 가중치 값에 미치는 영향을 분석하기 위하여 먼저, 모든 링크의 가중치를 0.1로 낮게 부여했을 때와 모든 링크의 가중치를 1.0으로 높게 부여했을 때, 그리고 5.2에서와 같이 링크의 의미에 따라 가중치 값을 다르게 부여했을 때의 문서 내 용어의 가중치가 링크 정보를 이용하지 않은 일반 벡터 검색 모델에서의 용어 가중치에 비해 얼만큼의 성능 향상을 가져오는지를 분석하였다. 전체 문서 집합을 대상으로 링크의 가중치 값에 따른 검색 조회율과 정확도를 분석한 결과는 <표 5>와 같다.

〈표 5〉 링크 가중치값에 따른 성능

링크 가중치	Recall(%)	Precision(%)
① 링크 무시	70.0	62.0
② all 0.1	74.0	63.3
③ all 1.0	76.0	61.7
④ 링크의 의미에 따라 부여	***90.0***	***73.8***

<표 5>의 ②에서처럼 모든 링크의 가중치를 0.1로 했을 경우에는 링크가 지시하는 원격 문서의 용어가 지역 문서의 용어의 빈도에 너무 낮게 계산되어 큰 성능 향상이 없었으며, ③의 경우에는 모든 링크의 가중치를 1.0으로 부여함으로써, ID가 6인 링크의 경우에 원격 문서에 있는 용어의 빈도가 너무 높게 측정되거나, 원격_원격 문서에 있는 용어도 너무 높게 측정되어, 관련 있는 문서가 문서 내 최대 용어 빈도 수에 의해 오히려 용어의 가중치가 낮게 계산되어진다. 그 결과, 검색 결과 집합의 총수는 많아지는데 반해 검색 결과 내 관련 문서의 수는 오히려 감소하여, 링크 정보를 이용하지 않은 ①의 경우에 비해 정확도가 더 낮게 나타남을 확인할 수 있었다. 그러나 ④의 경우처럼, 링크가 갖는 의미에 따라 가중치를 다르게 부여한 경우가 각각의 링크가 갖는 의미가 적절하게 반영되어 다른 링크 가중치 값에 비해 검색 조회율과 정확도가 더 높게 나타남을 확인할 수 있었다.

5.4 기존 하이퍼텍스트 검색 모델과의 비교, 분석

여기서는 제안된 모델과 기존의 하이퍼텍스트 검색 모델을 비교, 분석한다. 먼저 제안된 모델 즉, 위의 네 번째 모델을 'Scheme A'라 부른다. 제안한 모델을 두 개의 다른 모델과 비교 분석한다. 하나는 일반 벡터 검색 모델이다. 여기서는 이를 'Scheme B'라 부른다. 두 번째는 관련 연구 중에서 질의어 기반 검색에 링크의 방향성과 직접/간접 링크 정보를 활용하여 검색 결과 집합의 확장 및 문서의 순위를 변화시킨 모델이다. 이 모델을 'Scheme C'라 부른다. 전체 문서 집합에서 이 세 가지 모델을 이용하여 질의어에 대하여 문서의 조회율과 정확도를 조사하였다. 분석 결과는 <표 6>과 같다.

〈표 6〉 기존 검색 모델과의 비교

검색 모델	Recall(%)	Precision(%)
Scheme A	96.0	73.1
Scheme B	70.0	64.0
Scheme C	84.0	66.7

<표 6>의 결과에서, 검색 조회율의 경우, 제안된 모델이 일반 벡터 모델에 비해 26.0%, 링크 기반의 기존 모델에 비해 12.0% 향상되었음을 확인할 수 있다. 검색 정확도의 경우에도 일반 벡터 검색 모델에 비해 37.3%, 링크 기반의 기존 모델에 비해 23.0% 향상되었다. 이는 제안된 모델의 링크 의미 기반의 색인 과정에서는 용어와 관련 있는 문서의 경우는 문서내 용어의 가중치가 많이 높아지고, 관련 없는 문서의 경우는 용어 가중치가 현저히 낮아져서, 검색된 문헌의 총 수는 줄어드는 반면 검색된 관련 문헌의 수는 늘어남으로 인해서 기존 검색 모델과 비교해서 정확도가 더 향상됨을 확인할 수 있었다.

5.5 전체 문서 집합 크기를 고려한 정확도 분석

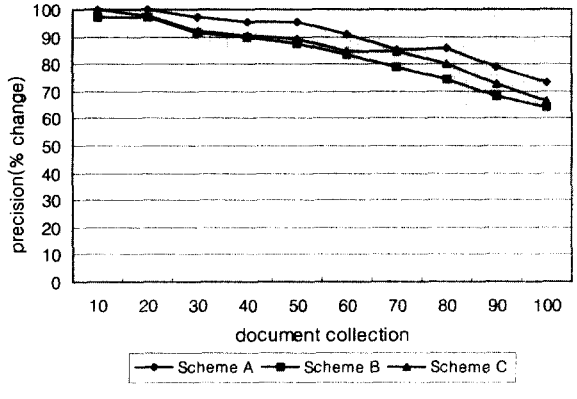
여기서는 전체 문서 집합의 크기가 검색 정확도에 미치는 영향을 분석하였다. 분석을 위하여 전체 문서 집합의 단위를 10으로 나누어 각 단위에서 기존 링크 기반 검색 모델과 일반 벡터 검색 모델과의 정확도를 분석하였다. 분석 결과는 <표 7>과 (그림 5)와 같다. <표 7>과 (그림 5)의 결과에서, 모든 단위의 문서 집합에서 링크 정보를 이용한 Scheme A와 Scheme C가 링크 정보를 이용하지 않은 Scheme B보다 성능이 향상됨을 확인 할 수 있고 전체 문서 집합의 크기가 커질수록 링크 정보를 이용하지 않은 것에 비해 링크 정보를 이용하는 것이 더 효율적임을 의미하며, 특히 전체 문서 집합의 크기가 커질수록 제안된 링크 의미 기반 검색 모델의 성능이 향상됨을 알 수 있다.

5.6 링크 메타 데이터 중심의 검색에 대한 분석

여기서는 제안된 모델에서 새롭게 정의하는 링크의 메타 데이터 중심으로 가장 많은 incoming link를 설 정 받은 문서 위주로 검색을 하는 것에 대한 효율을 분석하였다. 이를 위해 사용자의 질의어에 대해서 링크의 메타 데이터 중심의 검색과 문서의 내용 중심의 검색을 비교, 분석하였다. 실험을 위해서 먼저, 검색기 '포탈 알타비스타'와 '야후코리아'를 대상으로 질의어 'xml'을 검색해 본 결과 각 상위 10개의 순위에 해당되는 문서의 URL을 살펴보면 <표 8>과 같다. 이러한 순위는 정확한 검색 알고리즘은 알 수가 없지만 대체적으로 문서의 내용이 질의어와 얼마나 가까운지를 고려한 결과이다. 즉 문서 내용 중심의 검색 결과라고 볼 수 있다.

〈표 7〉 문서 집합의 크기에 따른 Precisio

Model / Result-set	Precision (%)		
	Scheme A	Scheme B	Scheme C
10	100.0	97.0	100.0
20	100.0	97.2	98.0
30	97.2	91.4	92.1
40	95.6	89.8	90.6
50	95.4	87.5	89.0
60	91.2	83.5	84.7
70	85.0	79.0	84.6
80	85.6	74.2	80.2
90	79.0	68.0	72.8
100	73.1	64.0	66.7



(그림 5) 검색 정확도에 대한 그래프

이러한 검색기 결과와는 상관없이 일반적으로 XML에 대한 좋은 정보를 가진 것으로 많이 알려진 사이트들을 살펴보면 <표 9>와 같다. 그러나 이러한 사이트들은 위의 2개의 검색기들의 검색 결과에서는 좋은 반응을 보이지 않았으며 어떤 사이트는 상위 순위에 포함되지도 않았음을 확인할 수 있다. 따라서 위의 결과에서 내용 위주가 아니라, 다른 사용자들에게 보다 인기 있는 사이트 위주로 검색을 하는 것도 필요하다. 이를 위해 질의어 'xml'을 메타 데이터로 갖는 inLinks의 가중치를 이용하여 LinkRSV를 계산하여 문서의 순위를 재조정하였다. 분석 결과는 <표 10>과 같다.

<표 10>의 결과에서 알 수 있듯이 [http : // www.w3.org/xml/] 사이트가 'xml'에 대한 메타 데이터로서 가장 큰 inLinks의 가중치를 가지고 있다. 즉 'xml'에 대한 메타 데이터로서 가장 많은 링크를 설정 받은 문서이다. 그러나 이 사이트는 위의 '야후 코리아' 검색기에서는 낮은 순위를 보였을 뿐만 아니라 '포탈 알타비스타' 에서는 상위 순위에 포함되지도 않았다.

결과적으로 보면, 정보 검색 결과는 질의를 던진 사용자의 주관적 판단 외에는 그렇게 큰 의미를 부여할 수 없다. 따라서 사용자의 질의에 가장 가까운 문서를 찾는 것은 결국에는 사용자의 판단에 의해서 이루어진다. 때문에 보다 다양한 방법으로 사용자의 욕구를 충족해 줄 수 있어야 한다. 따라서 질의어에 대하여 정보를 찾는 시각을 문서의 내용 중심의 검색에서 다른 방법의 검색까지 보다 다양화 할 필요가 있다. 이에 제안된 모델에서의 링크 메타 데이터 중심의 검색은 정보를 찾는 사용자에게 보다 다른 시각에서 원하는 정보에 접근 할 수 있는 새로운 방법을 제공할 수 있으며, 위의 분석 결과에서 그 필요성이 충분하다고 판단된다.

〈표 8〉 'xml'검색 결과 순위

순위	Yahoo	Altavista
①	www.xml.com/	www.xml.com
②	www.xml.org/	www.oasis-open.org/
③	www.ibm.com/developer/xml/	xml.com/
④	metalab.unc.edu/xml/	www.gca.org/conf/xml/
⑤	news : comp.text.xml	www.alphaworks-ibm.com/
⑥	www.w3.org/xml/	www.xmlinfo.com/
⑦	www.geovities.com/	webreview.com/wr/pub/XML/
⑧	metalab.unc.edu/pub/sun-info	cowu.jclark.com/xml/
⑨	www.sciam.com/	www.textuality.com/sgml-erb/
⑩	www.oasis-open.org/cover/	msdn.microsoft.com/

〈표 9〉 잘 알려진 XML사이트들

Top XML Sites
IBM's XML Web Site [www.ibm.com/developer/xml]
XML Developer Center [msdn.microsoft.com/xml/]
OASIS home page [www.oasis-open.org]
XML.org [www.xml.org]
The World Wide Web Consortium's XML [www.w3.org/xml/]
XML.COM [www.xml.com/xml/pub]
The XML Working Group FAQ [www.ucc.ie/xml/]

〈표 10〉 inLinks를 이용한 검색 순위

사 이 트	순위
http : //www.w3.org/xml/	①
http : //www.xml.com/	②
http : //www.xml.org/	③
http : //xml.com/	④
http : //www.ibm.com/developer/xml/	⑤
http : //msdn.microsoft.com/	⑥
http : //www.oasis-open.org/	⑦
http : //www.xmlinfo.com/	⑧
http : //www.textuality.com/sgml-erb/WD-xml.html	⑨
http : //metalab.unc.edu/xml/	⑩

6. 결 론

제안된 모델에서는 XML 링크를 검색에 활용하는 기법을 제시함에 있어서, 색인 시 링크 정보를 활용하는 메커니즘을 제시하였다. 먼저, 가상 문서의 개념을 이용하여 문서 내 용어 가중치를 계산함에 있어서 문서내 용어의 빈도 수를 재 정의하고 정규화 방법을 정의하였으며, 용어의 역 문헌 빈도 수를 재 정의하였다. 이를 위해 문서 내 링크의 유형과 행동에 관련된 속성을 기준으로 각 링크를 분류하여 식별자를 부여하고, 각 식별자마다 링크의 가중치를 부여하였다. 또한 다단계 링크나 순환적 링크를 제어하기 위한 새로운 속성을 정의하고 이 속성 하에서 다단계 링크의 가중치를 부여하였다. 다음으로 링크의 메타 데이터를 이용한 문서 내 inLinks의 가중치를 계산하여 메타 데이터 중심의 검색을 가능하게 하였다.

제안된 모델에 대한 실험 및 분석 결과에서 링크의 의미에 따라 검색 성능이 향상됨을 확인하였으며, 검색 결과 집합에 포함된 문서들에만 링크 정보를 활용한 기존의 검색 모델에 비해서도 성능이 향상됨을 확인할 수 있었다. 또한 전체 문서 집합의 크기가 커질수록 제안된 모델의 성능이 향상됨을 확인할 수 있었다. 마지막으로 링크의 메타 데이터 중심의 검색 결과에서 검색 시각의 다양화가 필요함을 확인할 수가 있었다.

앞으로의 연구 과제로는 대규모 XML 문서 집합을 대상으로 XLink을 이용한 검색 모델의 효율성을 검증하고 아울러 문서의 논리적 구조 정보를 이용한 문서의 부분 검색 기능과의 통합을 통해서 XML 문서 검색 시스템을 개발하는 것이다.

참 고 문 헌

[1] Cathal Gurrin & Alan F. Smeaton, "A Connectivity Analysis Approach to Increasing Precision in Retrieval from Hyperlinked Documents," cgurrin@ compapp.dcu.ie, 1999.

[2] Dario Lucarella "A model for hypertext- based information retrival," In Hypertext Concepts systems, and Applications, Eds. Rizk, Streitz, and Andrie, 1990.

[3] Frei, H. P., & Stieger, D., "The use of semantic links in hypertext information retrieval," Information Processing& Management, Vol.31, No.1, pp.1-13, 1995.

[4] Jacques Savoy, "An extended Vector-processing scheme for searching information in hypertext systems," Information processing & Management, Vol.32, No.2, pp.155-170, 1996.

[5] Philippe Martin and Peter Eklund, "A Key for Enhanced Hypertext Functionality andVirtual Documents," http : // www.cs.unibo.it/~fabio/ VD99/ecklund/.

[6] Sylvie Ranwez and Michel Crampes, "Conceptual Documents and Hypertext Documents are two Different Forms of Virtual Document," http : //www.cs.unibo.it/~fabio/VD99/ ranwez/.

[7] W3C Candidate Recommendation 3 July 2000, "XML Linking Language(XLink) Version1.0," http : //www.w3.org/TR/2000/CR-xlink-20000703/.

[8] W3C Candidate Recommendation 3 July 2000, "XML Pointer Language(XPointer) Version1.0," http : //www.w3.org/TR/2000/CR-xptr-20000703/.

[9] Savoy, J., "Retrieval effectiveness of information retrieval systems used in a hypertext environment," Hypermedia, Vol.5, No.1, pp.23-46, 1993.

[10] Savoy, J., "A learning scheme for information retrieval in hypertext," Information Processing& Management, Vol.30, No.4, pp.515-533, 1994.

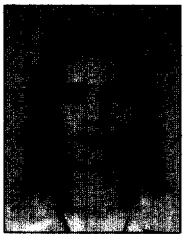
[11] Savoy,J., "Ranking schemes in hybrid boolean systems : A new approach," Submitted, 1996.

[12] Hee-Yeol Ryu and Eun-Jung Kim and Jong-Min Bae, "A Term Weighting and Ranking Scheme Using Hypertext Links," Proceedings of The 1st International Conference on East-Asian Language Processing and Internet Information Technology, 2000.

[13] 김동욱, 류준형, 주원균, 맹성현, "링크 정보를 이용한 검색 신뢰도의 향상", 한국정보과학회 봄 학술발표논문집, Vol.25, No.1, 1998.

[14] 김상준, 김은정, 배종민, "XML 링크의 메타 데이터를 이용한 검색 시스템의 설계", 한국정보과학회 봄 학술발표논문집 Vol.27, No.1, 2000.

[15] 김은정, 배종민, " XML 링크 정보를 이용한 정보 검색 색인 기법의 설계", 한국정보처리학회논문지, 제 7권 제 7호, 2000.



김 은 정

e-mail : ejkim@taejo.pufs.ac.kr

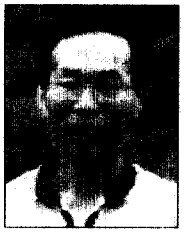
1989년~1993년 LG전자 영상미디어 연구소 연구원

1996년 경상대학교 대학원 전자계산학과 졸업(공학석사)

2001년 경상대학교 대학원 전자계산학과 졸업(공학박사)

2000년~현재 부산외국어대학교 전자컴퓨터공학부 강의전담 전임강사

관심분야 : 정보검색, 디지털 라이브러리, XML 데이 터베이스 통합



배 종 민

e-mail : jmbae@base.gsnu.ac.kr

1980년 서울대학교 사범대학 수학교육과 (학사)

1983년 서울대학교 대학원 계산통계학과 (석사)

1995년 서울대학교 대학원 계산통계학과 (박사)

1982년~1984년 한국전자통신연구소 연구원

1997년~1998년 Virginia Tech. 객원연구원

1984년~현재 경상대학교 전임강사, 조교수, 부교수, 교수

관심분야 : XML 데이터베이스통합, 정보검색, 디지털 라이브러리, 데이터마이닝