노드정보를 이용한 문서검색의 성능에 관한 연구

A Study on the Performance of Structured Document Retrieval Using Node Information

윤소영(Soyoung Yoon)*

초 록

노드는 문서를 구성하는 작은 크기의 의미있는 정보 단위이다. 정보검색에 문서의 구조정보를 이용함과 더불어 문서보다 작은 검색단위에 대한 연구가 활발히 이루어지고 있다. 이 연구에서는 노드정보를 이용한 검색실험을 위해 벡터공간모델 검색기법을 사용하여 다양한 유사도 산출방식을 적용한 실험과 구조정보를 활용한 확장 실험을 수행하였다. 실험결과 문서의 유사도를 산출하는 방식에 따른 검색성능의 차이는 거의 나타나지 않았으며, 구조정보를 적용하는 확장 노드검색이 가장 좋은 성능을 나타냈다.

ABSTRACT

Node is the semantic unit and a part of structured document. Information retrieval from structured documents offers an opportunity to go subdivided below the document level in search of relevant information, making any element in an structured document a retrievable unit. The node-based document retrieval constitutes several similarity calculating methods and the extended node retrieval method using structure information. Retrieval performance is hardly influenced by the methods for determining document similarity. The extended node method outperformed the others as a whole.

키워드 : 노드정보, 구조검색, XML 검색, 벡터공간모델, 확장 벡터공간모델
hierarchical information, structural retrieval, XML retrieval, VSM(Vector Space Model), extended VSM

* 국사편찬위원회 사료연구위원 (syoon@history.go.kr)
■ 논문접수일자 : 2007년 2월 15일
■ 게재확정일자 : 2007년 3월 12일

1. 서 론

전통적인 정보검색의 방식은 정보 요구가 발생했을 때 전체 문서를 검색하여 결과를 보여주는 반면에, 최근 정보검색 시스템은 문서의 구조와 검색결과 측면에 제한을 두는 더 세분화된 검색 패러다임을 구현하고 있다. 즉 전체 문서를 검색하는 것이 아니라 이용자의 정보요구에 부응하면서도 문서보다 작은 크기의 특정 부분을 검색하려는 목표를 가지고 있다. FERMI 멀티미디어 정보 검색 모델에 따르면 이러한 문서의 특정 부분은 정보 요구를 만족시킬 정도로 충분하면서 문서구조상의 가장 특정적인 요소가 되어야 한다고 했다(Chiaramella, Mulhem, and Fourel 1996). 이러한 방법으로 정보검색이 수행되면 이용자는 더 작은 단위, 즉 문서의 일부를 검색결과로 얻을 수 있게 되어 검색 결과를 살펴봐야 하는 시간과 수고를 줄일 수 있게 된다.

현재 XML은 인터넷을 포함한 다양한 장소에서 데이터 교환의 실질적인 표준으로 자리잡고 있으며 산업계뿐만 아니라 학계를 포함한 거의 모든 분야에서 각종 데이터가 XML로 저장, 교환되고 있다. XML 마크업의 주목적이 문서의 논리적 구조의 명확한 표현이라고 할 때, 정보검색 측면에서는 XML 문서의 구조정보를 이용한 정보검색 성능의 향상을 기대할 수 있다.

XML 문서의 구조정보를 이용한 정보검색 실험 이전에도, 문서의 구조를 이용하여 문서의 일부인 섹션이나 문단 등의 단락(passage)을 검색단위로 하여 정보검색의 성능을 향상시키고자 한 연구들이 있었다(Salton, Allan, and Buckley 1993; Wilkinson 1994). 문서의 물리적 구조에 기반하여 선형적으로 문서를 나누어 검색을 수행하였을 경우에는 문서검색보다 오히려 성능이 좋지 않았으나(Wilkinson 1994), 구조정보를 추가정보로 사용하였을 경우에는 검색성능이 향상되는 결과를 보여주었다(Salton, Allan, and Buckley 1993).

이 연구에서는 정보검색의 대상을 문서가 아닌 XML 노드로 정하고, 벡터공간모델(VSM; Vector Space Model) 검색기법을 적용하여, 세 가지의 문서 유사도 산출방식에 따른 노드검색 실험과 계층적 구조정보를 추가하는 확장 노드검색 실험을 수행한 후 각각의 검색성능을 비교, 평가하였다.

2. 구조문서의 검색

2.1 검색단위

정보검색은 계층적 내부 구조를 가지는 문서의 검색이라는 관점에서 볼 수 있는데, 예를 들어 문서는 섹션(section), 문단(paragraph) 그리고 문장(sentence)을 가지며 각 문서 구조의 요소들은 검색에 이용될 수 있는 검색단위가 될 수 있다(Callan 1994). FERMI (Formalisation and Experimentation on the Retrieval of Multimedia Information) 모델에서는 문서를 구조적인 측면에서 분석하여 트리구조로 표현하고 트리의 수준(level)을 구분하여 검색단위가 될 수 있는 요소를 그룹화하여 표현하고 있다(Chiaramella, Mulhem,

and Fourel 1996).

Hanato 등(2002)은 적절한 XML 검색단위를 결정하기 위해 XPath를 응용한 데이터 모델을 정의하고 검색 실험을 수행하였다. 검색 결과 단위로서 적합한 노드를 CPD(Coherent Partial Document)라 정의하고 그 크기를 알아내기 위해 XML 문서의 문서구조에 대한 색인파일과 XML 노드의 단어빈도에 기반한 색인파일을 생성한 후, 통계정보(R=대상단어의 종수/단어 총수)를 추가하였다. 그 결과, XML 문서를 CPD로 나누어 XML 노드를 효과적으로 검색할 수 있음을 밝혔다.

Kamps 등(2003)은 XML 검색에서 검색단위에 관심을 가지고 두 가지의 색인 파일을 만들어 검색 실험을 수행하였다. 하나는 문서를 대상으로 하는 색인파일이고, 다른 하나의 색인 파일은 XML 노드와 잠재적으로 검색가능한 모든 XML 노드를 대상으로 생성하였다. 그는 내용에 대한 질의만을 대상으로 검색 실험을 수행하였는데 실험 결과 이용자나 평가자 모두 XML 컬렉션의 의미 있는 검색단위로 전문(full-text)에 가까운 섹션 정도의 검색단위가 적절하다고 평가했다. 후속 실험에서 검색 대상이 되는 노드간의 길이의 편차가 심하여 검색에 영향을 미치므로 기준값을 적용해 노드 길이를 정규화하는 실험을 수행하여 검색성능이 향상됨을 보여주었다(Kamps, Rijke, and Sigurbjönsson 2004).

문서보다 작은 단위의 하위 구성요소를 검색하려는 시도는 기존 정보검색의 단락검색(passage retrieval)에서 이미 찾아볼 수 있다. 문서의 일부를 검색한다는 의미로 보면 노드를 대상으로 하는 검색도 단락검색에 포함되는 것으로 볼 수 있다. 단지 차이라고 한다면 XML 문서검색에서는 단락을 구분하는 방법으로 문서 트리구조의 노드를 그대로 이용한다는 것이다.

단락검색은 검색의 결과를 단락으로 제시할 수도 있고, 전체 문서로 보여줄 수도 있다. 단락검색의 결과를 문서로 제시할 경우에는 단락과 질의간의 유사도를 기반으로 문서와 질의간의 유사도를 결정해야 하며 이를 바탕으로 문서의 순위를 매겨야 한다. 문서의 순위를 매기는 가장 간단한 방법은 문서를 구성하는 단락들 중에서 질의와 가장 높은 유사도를 가진 단락의 유사도를 문서의 유사도로 간주하는 것이다. 다른 방법으로는 각 단락의 유형에 따른 가중치를 미리 정해 놓고 질의와 단락간의 유사도와 단락 유형에 따른 가중치를 곱해서 나온 값에서 가장 높은 값을 가지는 단락의 유사도를 문서의 유사도로 간주하기도 한다. 이외에 문서를 구성하는 단락들의 유사도를 다 합한 다음에 단락의 수로 나누어 주는 방법과 문서를 구성하는 단락들을 유사도 순으로 정렬한 다음에 유사도들을 합할 때 순위가 낮아질수록 낮은 가중치를 주어 합하는 방법이 있다(Wilkinson 1994). 문서검색과 단락검색을 합쳐서 문서의 순위를 매기는 방법도 있는데 우선 단락검색을 하기 전에 문서검색을 통해 각 문서의 유사도를 구한 다음 단락검색을 실행한다. 그 다음에 문서를 구성하는 단락들 중 가장 높은 유사도를 얻은 단락의 유사도를 문서의 유사도로 간주하여 이 유사도와 문서 검색에서 얻은 유사도를 합하여 문서의 순위를 매기는 것이다(Wilkinson 1994; Callan 1994).

Salton 등(1993)은 단락(passage)을 검색단위로 하여 검색 실험을 수행하고 문단과 문단의 집합인 섹션 그리고 더 작은 단락인 문장 등을 검색에 활용하면 문서검색보다는 검색성능이 향상될 수 있음을 보여주었다. Wilkinson (1994)은 구조화 문서에 대한 검색 실험에서 문서의 순위리스트에서 구조를 형성하는 단락을 추출하는 것은 좋지 않다는 결론을 내렸으나, 문서의 구조가 문서검색 성능을 향상시킬 수 있다는 사실을 알아냈다. 그는 문서단위 검색 실험, 문서의 섹션을 이용하여 단락검색을 수행한 실험, 그리고 각 섹션의 유형에 따라 가중치를 부여하는 실험 등 세 가지를 기본으로 하여 각 코사인 유사계수를 여러 형식으로 조합하거나 섹션의 길이를 일정하게 고정하는 등 여러 방식으로 검색 실험을 수행하였다.

2.2 벡터공간모델 검색

벡터공간모델(Vector Space Model) 검색 시스템에서 문서의 순위를 매기는 방법은 문서와 질의를 용어 벡터로 처리하는 벡터공간 모델에 기초하여 수행된다. 각 용어의 가중치는 문서와 질의의 단어빈도(term frequency)에 비례하고 용어가 나타나는 문서의 총수인 문헌빈도(document frequency)에는 반비례하게 부여된다. 문서와 질의간의 유사도는 일반적으로 코사인 유사도로 측정된 두 벡터간의 거리로 정의된다.

Mass 등(2002)과 Carmel 등(2003)은 XML 문서를 구조화되지 않은 텍스트 문서와 완벽한 구조를 가진 데이터베이스 데이터간의 중간 경계쯤에 위치하는 것으로 규정하였다. 기존의 정보검색기법은 구조화되지 않은 문서를 대상으로 검색을 제공하는 반면, 데이터베이스 기법은 완벽한 구조를 가진 데이터를 다루고 있다고 보았다. 기존에 수행되었던 XML 질의나 검색에 대한 연구들이 데이터베이스 구축을 중심으로 한 구조적인 측면에 근거를 두고 수행한 것과는 다르게, Mass 등(2002)의 연구에서는 정보검색관점에서 질의 처리나 검색결과의 순위화 등으로 검색효율성을 평가하였다. 확장 벡터공간모델을 이용하였으며, 용어가중치로 $tf \cdot idf$ 가중치를 사용하였다.

확장 벡터공간모델 검색에서는 색인 단위로서 용어와 그 용어가 나타나는 XML 구조의 쌍인(t, c)을 사용한다. 용어가 나타나는 XML 구조 c(context)를 표현하기 위해 XPath를 모델을 이용하여 (t, c) = (search, /book/title/) 과 같은 형식으로 표현하였다. 순위화 알고리즘으로는 다음과 같이 코사인 유사계수를 변형하여 사용하였다.

$$\rho(Q,D)=\frac{\sum_{(t_i,c_j)\in q}\sum_{(t_i,c_j)\in d} W_Q(t_i,c_j)*W_D(t_i,c_k)*cr(c_j,c_k)}{\|Q\|*\|D\|}$$

Q : 질의

D : 문서

$W_Q(t_i, c_j)$: 질의에 나타나는 (용어, 구조) 쌍

$W_D(t_i, c_k)$: 문서에 나타나는 (용어, 구조) 쌍

$cr(c_j, c_k)$: 문맥유사도(context resemblancce)

예를 들어 질의가 (John, /author)은 문서의 (John, /fm/author/fnm)와 (John, /bm/author/fnm)를 검색하게 된다. 이때 $cr()$값은 질의와 문서간 구조표현의 일치 정도를 나타내며 0에서 1 사이의 값을 갖는다. 질

의와 문서의 구조가 정확히 일치하면 1이 되고 질의의 구조(/author)가 문서 구조(/fm/author/fnm)와 길이에 차이가 있을 경우 값은 루트로부터 시작한 길이의 차이가 되어 $cr()$=(/author, /fm/author/fnm) = 2/4 = 0.5가 된다.

문서 전체가 아닌 XML 노드의 순위를 정하기 위해서는 전통적인 벡터공간모델을 노드, 즉 문서의 구성요소 수준에서 통계값을 구하도록 확장시켜야 한다. 문제는 XML 문서의 구성요소가 중첩된 포함관계를 가지므로 용어의 발생빈도를 구할 때 문서의 계층관계를 고려해야 한다는 것이다. 더 명확히 말하면 특정 용어가 한번이상 계산되면 안 된다는 것이다. 예를 들어 문단에 나타나는 용어는 그 문단을 포함하는 섹션에서도 나타나게 된다. 이 때 용어의 노드 빈도는 무엇이 되는가의 문제가 발생하게 된다. 두 노드에 속하는 것으로 계산하게 되면 실제로는 문서에 한번만 나타나기 때문에 순위가 왜곡될 수 있다. 반면에 한번만 나타는 것으로 계산한다면 어느 노드에 나타나는 것으로 계산해야 하는가 하는 문제가 발생하게 된다.

이 연구에서는 이러한 두 가지 문제점을 해결하기 위하여 XML 문서의 색인 작성 방법에 따라 원칙을 정하였다. 노드검색에서는 기존의 단락검색과 같이 포함관계를 가지는 노드의 중복색인을 허용하고 용어가 발생하는 노드마다 빈도를 각각 계산하여 부여하게 하였다. 반면에 구조의 계층정보를 추가하는 확장 노드검색은 독립색인 방식으로 XML 문서의 색인을 구축하고 통계의 이벤트 확률을 구할 때 이용하는 확장인자(augment factor)를 적용하여 상위 노드의 가중치를 재계산하여 이 문제를 해결하였다.

3. 실험설계

3.1 실험데이터

실험문서집단은 학위논문이나 학술지와 같이 논리적 계층 구조를 가진 문서의 전문(full text)이며 XML 구조로 표현되어 있어야 한다. 이 실험에서는 실험환경에 맞는 실험집단을 구축하기 위해 ACM이 출판하는 학술회의 논문집 중에서 1998년에서 2004년 사이에 출판된 정보검색 및 정보기술 관련 주제를 다루고 있는 SIGIR 281건, SIGMOD 396건, CIKM 228건, 그리고 ACM DL 125건 등 총 1030건을 수집하였다. 수집한 문서를 XML 문서로 변환하기 위해 문서의 논리적인 구조를 분석하였으며 그 결과에 따라 XML 문서를 생성하였다.

실험에서 사용한 질의는 실험대상 문서의 내용을 분석하여 30개를 추출하였으며 질의 당 평균 단어 수는 5개이다. 질의에 대한 문서의 적합성 평가는 정보학 전공 대학원생 4명이 〈표 1〉에서 보이는 질의 30개에 대해 실험대상 문서 1030건에 대해 적합성을 미리 판정하도록 하여 4명 중 3명 이상이 적합하다고 판정한 문서를 적합하다고 간주하여 질의의 적합문서를 구성하였다. 실험의 성능 평가는 재현율 0에서부터 1까지 0.1씩 건너뛴 11지점에서의 정확률 평균값인 11-지점 평균정확률(11-point average precision), 검색결과 상위

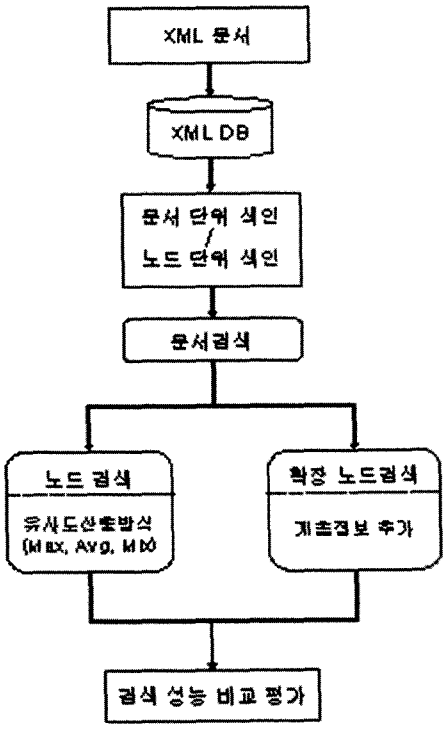
10위에서의 정확률(p(10)), 그리고 상위 20위에 서의 정확률(p(20))을 기준 척도로 사용하였다.

〈표 1〉 실험에서 사용된 질의 집단

번호	질의	번호	질의
1	content-based image retrieval system technique	16	multimedia spatial access methods
2	multimedia information retrieval system reference database	17	feature selection algorithms methods
3	recommender system user interests preference personalized	18	user relevance feedback techniques
4	singular value decomposition (SVD)	19	structured document retrieval hierarchical
5	information data visualization	20	automatic speech recognition algorithms technique
6	concurrency control semantic transaction management application performance benefit	21	k-nearest neighbor(knn) method classifier
7	machine learning algorithms adaptative probabilistic model	22	distributed database information keyword based queries search
8	user evaluation retrieval efficiency	23	neural network algorithms model
9	probabilistic latent semantic indexing analysis	24	query expansion term re-weighting
10	support vector machine(SVM)	25	citation indexing retrieval links
11	question answering system user interface	26	text summarization information extraction retrieval
12	knowledge discovery data mining web KDD association rule	27	pattern recognition discovery matching algorithms
13	bayesian network model inference naive decision theory	28	spoken document retrieval
14	information filtering methods algorithms applications statistical retrieval models	29	natural language retrieval
15	xml documents structure retrieval query languages	30	browsing document using hypertext model

3.2 실험개요

〈그림 1〉은 노드정보를 이용하는 문서검색에 벡터공간모델 검색기법을 적용하여 검색실험을 수행하는 전체적인 실험과정을 보여주고 있다. 전통적인 정보검색 모형은 검색 최소 단위가 문서인 반면, XML 마크업은 문서를 트리구조로 가지고 있어서 검색시 질의에 적합한 작은 크기의 서브트리를 탐색하게 된다. 따라서 대부분의 XML 검색기법들은 색인 노드라고 불리는 개념에 기반하여 검색을 수행하고 있다. 검색시 모든 XML 노드가 검색단위가 될 수는 없는데, 그 이유는 노드가 너무 세분화되어 있거나 섹션 제목이 없는 섹션 본문과 같이 중요한 노드를 빠뜨린 경우가 있기 때문이다. 그래서 먼저 검색대상이 될 색인 노드의 집합이 정의되어야 하는데, 그 방법으로는 DTD에 기반하여 정의하는 방법과 경험에 의해 문서마다 특정 색인 노드를 정의하는 방법이 있다. 색인 노드의 정의를 위해 다음과 같은 두 가지의 접근방법을 고려해 볼 수 있다.(Abolhassani and Fuhr 2004). 이를 바탕으로 본 실험에 앞서 색인 서브트리(indexing subtrees)에 기반한 중복색인 작성방법과 독립 단위(disjoint units)에 기반한 독립색인 작성방법의 두 가지 색인 방식을 적용한 예비실



〈그림 1〉 노드검색 실험과정

험을 수행하였다. 이 두 가지 접근방법은 색인 가중치를 구하는데 있어서도 차이를 보이는데 먼저 색인 서브트리 방식에서는 단어빈도, 문서 길이 등을 먼저 구한 후 단어가중치는 단어빈도 합과 문서길이 합의 함수로부터 구하게 된다. 반면에 독립 색인 방식에서는 먼저 말단 노드의 색인 가중치를 계산한 후, 상위 또는 내부 색인 노드(inner nodes)의 가중치는 그 말단 노드의 가중치를 조합하여 유도한다.

확장 노드검색 실험에서는 색인 간의 중복을 피하기 위해 독립색인(disjoint units) 방식으로 구축된 노드 색인을 이용하였다. 독립색인 방식은 색인의 중복을 피하기 위해 노드를 독립되어 있다고 간주하여 작성된다. 그러나 색인 노드는 여러 하위 노드로 구성되어 있으므로 그 색인 노드의 용어가중치는 하위 노드의 용어가중치와 어떤 방식으로든 결합되어 재계산하여야 하는 문제가 발생하게 된다. 이를 위하여 동일한 용어가 색인 노드에 포함된 하위노드에도 출현하였을 경우에 색인 단위에서 확장인자를 이용하여 용어가중치를 계산하도록 하였다. 예를 들어 섹션은 여러 서브섹션으로 구성이 되지만 섹션에서 서브섹션 노드가 제외되고 색인이 생성되므로 동일한 용어가 서브섹션을 제외한 섹션 노드와 각 서브섹션 노드마다 다른 가중치를 가지게 된다. 따라서 섹션의 용어가중치는 서브섹션의 용어가중치와 결합하여 계산된 값으로 재설정되어야 한다. 이를 위해 상하 계층에 공통적으로 나타나는 용어를 하위 노드를 중심으로 색인한 후 상위 노드에 나타나는 용어의 가중치를 계산하는 방법을 사용하였다.

(1) 문서검색

문서검색 실험에서는 문서검색을 수행하여 다른 검색기법들과 비교하기 위한 기초 검색결과로 사용하였다. 적용된 가중치는 단어빈도와 역문헌빈도의 곱($tf*idf$), 그리고 단어빈도에 로그값을 취한 값과 역문헌빈도의 곱($ltf*idf$)이다. 질의 x와 문서 y간의 유사도는 코사인 유사계수를 이용하였다.

$$tf*idf = tf*(1+\log_2 \frac{N}{df})$$

$$tf*idf = (1+\log tf)*(1+\log_2 \frac{N}{df})$$

N : 문서집단의 문서총수
df : 문서빈도

$$\text{코사인 유사도(x, y)} = \frac{\sum(x_i \cdot y_i)}{\sqrt{\sum(x_i)^2 \cdot \sum(y_i)^2}}$$

(2) 노드검색

XML 문서검색에서는 노드가 검색단위가 되므로 노드 각각을 색인 단위로 간주하여 색인 서브트리 방식에 의한 중복색인을 작성하였다. 각 노드를 위해서 노드의 모든 텍스트와 노드에 포함된 하위 노드의 텍스트 모두를 대상으로 색인을 작성하였다. 다시 말하면 XML 노드를 기본 색인 단위로 하여 텍스트만을 포함하는 최하위의 말단 노드를 제외한 상위의 노드들, 예를 들어 섹션은 여러 서브섹션과 문단 등의 하위 노드를 포함하여 색인이 작성된다. 색인 작성 방식으로 보면 이 실험에서의 노드 색인은 중복색인을 하는 색인 서브트리 방식의 색인이다.

XML 문서검색에서 검색 대상은 기본적으

로 XML 문서의 텍스트 부분으로 XML 트리의 말단 노드만이 대상이 된다. 이러한 말단 노드가 검색결과로 제시된다면 검색결과가 한 단어 또는 한 문장 등으로 너무 세분되어 검색 효율성 측면에서 의미 있는 검색결과를 기대하기 어려우며, 또한 검색대상 노드들이 너무 많아 시스템의 성능도 기대할 수 없게 된다(Gövert et al 2002). 이러한 측면에서 검색의 단위는 FERMI 멀티미디어 모델의 구조적 관점을 따른 색인 노드의 개념에 기반하는 것이 타당하다고 볼 수 있다. 이러한 XML 문서검색의 검색단위에 대한 논의는 노드를 어떠한 기준으로 나누어 줄 것인가 하는 문제와 동일선상에 놓이게 된다.

노드검색 실험에서 사용한 가중치는 문서의 용어가중치를 노드 기반 XML 문서검색에 맞도록 적용한 노드내 단어빈도의 로그값($ltfe$), 로그단어빈도*역노드빈도($ltfe*ief$)를 사용하였다(Wolff, Flörke and Cremers 2000). 노드는 문서보다 상대적으로 길이가 짧아 노드내 단어빈도로 1이 많이 나타나게 되므로, 단어빈도가 1인 용어의 낮은 영향력을 보완하도록 로그 단어빈도($ltfe$)를 이용하였다. 노드검색 실험에 사용한 가중치 공식은 다음과 같다.

$$ltfe = 1 + \log(tfe)$$

$$ltfe*ief = (1 + \log(tfe))*(1 + \log_2 \frac{NE}{ef})$$

tfe : 노드내 단어빈도
NE : 문서내 노드 총수
ef : 노드 빈도

문서의 유사도를 산출하는 방법은 가장 높은 노드의 유사도를 문서의 유사도 산출하는 방법(최대값)과 노드의 유사도를 모두 합하여 노드의 수로 나눈 값, 즉 노드의 유사도 평균을 문서의 유사도로 산출하는 방법(평균값), 그리고 노드의 유사도를 문서의 유사도와 혼합하여 문서의 유사도로 산출하는 방법을 이용하여 성능을 평가하였다. 유사도를 혼합하는 방법에서는 가장 높은 유사도를 얻은 노드의 유사도를 문서의 유사도로 간주한 다음 이 유사도와 문서검색에서 얻은 유사도를 합하여 문서의 유사도로 산출하였다.

(3) 확장 노드검색

가중치 관점에서 보면 XML 노드는 하나의 문서처럼 독립적으로 처리되어야 색인어의 단어빈도, 문서빈도 등을 한번만 산출할 수 있게 된다. 또한 계층구조를 가지는 문서를 대상으로 검색을 수행할 때 일부의 질의에 대해서만 하위 계층에 나타나는 노드를 적합한 결과로 제시하고, 일반적으로는 트리 구조상 최상위에 해당하는 문서 전체를 가장 적합한 결과로 검색하게 된다. 그 이유는 여러 계층에 공통적으로 나오는 용어에 가중치를 부여하는 방식에서 찾을 수 있다. 일반적으로 계층 구조를 가지는 검색을 수행할 때 색인어가 여러 계층에 공통으로 나타날 때 하위 노드의 가중치보다 상위 노드의 가중치를 더 높게 부여하기 때문에 계층 구조상 상위의 노드나 문서를 적합한 검색결과로 제시하게 되는 것이다(Gövert, et al. 2002).

이 실험에서는 이러한 두 가지 측면을 모두 수용하기 위하여 계층 구조를 가진 문서의 색인어에 대한 용어가중치를 계산하기 위한 출발점을 가장 특정적인 하위 노드인 문단에서부터

시작하였다. 트리구조에서 보면 상위 수준의 노드는 하위의 다른 노드들의 집합으로 구성될 수 있으므로 다른 노드를 포함하지 않는 가장 하위의 텍스트 노드만을 색인하면 된다. 이러한 색인 노드의 개념을 이용하면 색인 노드 트리 구조가 최종에는 문서를 검색하는 결과가 된다. 예를 들어 서브섹션, 섹션, 논문을 노드로 정의한다고 하면 독립적인 색인단위는 하위 노드를 제외한 그 노드가 될 것이다.

독립색인 방식으로 작성된 노드 색인에서 〈그림 2〉의 예에서처럼 특정 용어 XPath가 섹션과 서브섹션 노드에 동시에 나타나게 되면 상위 색인 노드인 섹션의 XPath 용어가중치를 재계산하여야 한다. 이때 용어가중치를 조합하는 방법은 일반적으로 두 용어가중치를 더하여 사용하게 된다. 이렇게 하면 섹션의 용어가중치가 항상 큰 값을 갖게 되어 XPath를

$$W = W_i + \lambda \cdot W_j - W_i \cdot \lambda \cdot W_j$$

W_i : 상위 노드의 가중치
λ : 확장인자
W_j : 하위 노드의 가중치

포함하는 질의로 검색을 실행하였을 때 상위 순위로 섹션을 검색하게 되고 두 번째 서브섹션은 결과에서 낮은 순위로 나타나게 된다. 즉 하나의 용어에 대해 여러 가중치를 더하게 되면 가장 일반적인 상위 노드에 가장 높은 가중치를 부여하게 되어 수준이 서로 다른 검색단위를 검색결과로 허용하지 않게 된다. 이러한 모순된 가중치 조합 방법을 보완하기 위하여 이 실험에서는 Fuhr 등(1998)이 DOLORES 멀티미디어 시스템에서 확률규칙을 이용하여 제안한 확장(augmentation)이라는 개념을 채택하였다. 다음의 용어가중치 공식은

XML 문서
<section> 0.3 XPath
<subsection> 0.5 example </subsection>
<subsection> 0.8 XPath 0.7 syntax </subsection>
</section>

* 가중치를 더하여 재부여된 XML 문서
<section> 1.1 XPath
<subsection> 0.5 example </subsection>
<subsection> 0.8 XPath 0.7 syntax </subsection>
</section>

** 확장인자 0.4을 이용하여 가중치가 재부여된 XML 문서
<section> 0.524 XPath
<subsection> 0.5 example </subsection>
<subsection> 0.8 XPath 0.7 syntax </subsection>
</section>

〈그림 2〉 확장 노드검색의 용어가중치의 예

Bilingsley(1979)가 제안한 이벤트를 위한 확장 가중치 공식을 이 실험에 맞도록 응용한 공식이다.

예를 들어 〈그림 2〉에서 XPath에 대한 섹션의 가중치 0.3과 서브섹션의 가중치 0.8이 있을 때 확장인자(augmentation factor) 0.4를 사용하면 섹션의 XPath의 가중치를 0.3 + 0.4 · 0.8 − 0.3 · 0.4 · 0.8 = 0.524로 재부여 하게 되어 서브섹션의 0.8이 순위에서 더 앞서게 하는 결과가 된다.

4. 실험 결과 분석 및 평가

4.1 노드검색 실험 결과

노드를 이용한 문서검색 실험에 앞서, 실험집단에 대해 문서검색 실험을 수행하여 검색성능 비교를 위한 기초 결과로 이용하였다. 문서검색 실험 결과를 보면 단어빈도에 로그값을 취한 로그단어빈도*역문헌빈도 가중치의 검색성능이 가장 좋게 나타났다. 그 이유는 문서의 길이에 영향을 받기 때문에 단어빈도에 로그값을 취해 고빈도어의 영향력을 감소시킨 결과가 검색성능을 향상시킨 것으로 보인다.

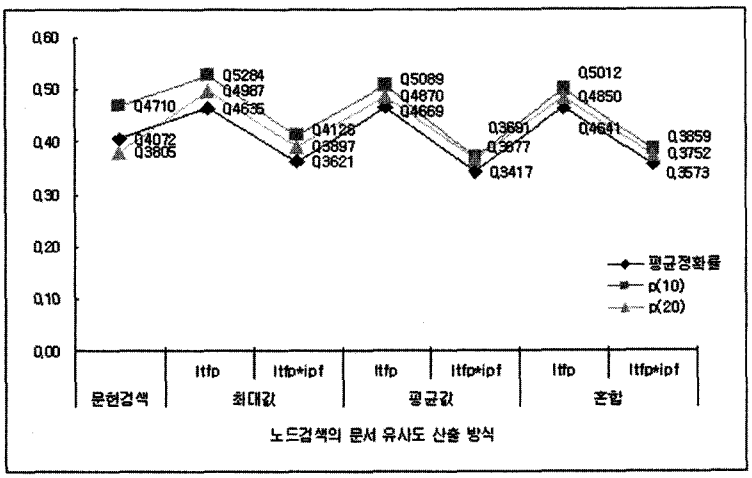
노드검색 실험에서는 XML 구조에 기반하여 노드를 구분하여 노드검색을 수행한 결과를 비교 평가하였다. 노드검색에 적용한 용어가중치는 로그 단어빈도(*ltfe*), 그리고 로그 단어빈도*역노드빈도 (*ltfe*∗*ief*)이다. 노드검색에서 문서의 유사도를 산출하는 방법은 가장 높은 노드의 유사도를 문서의 유사도 산출하는 방법(최대값)과 노드의 유사도를 모두 합하여 노드의 수로 나눈 값, 즉 노드의 유사도 평균을 문서의 유사도로 산출하는 방법(평균값), 그리고 노드의 유사도를 문서의 유사도와 혼합하여 문서의 유사도로 산출하는 방법을 이용하였다. 노드의 유사도와 문서의 유사도를 혼합하는 방법에서는 가장 높은 유사도를 얻은 노드의 유사도와 문서검색에서 얻은 유사도를 합하여 문서의 유사도로 간주하였다.

사전 실험을 통해 알아본 바에 의하면 적용한 가중치 중 단어빈도(*tfe*)보다는 로그 단어빈도(*ltfe*)가 더 좋은 성능을 나타냈다. 그 이유는 노드가 문서에 비해 상대적으로 길이가 짧아 단어빈도로 1이 많이 나타나므로 로그값으로 이러한 저빈도 단어들의 영향력을 보완하도록 한 로그 단어빈도(*ltfe*)가 더 나은 성능을 보인 것으로 분석된다. 역노드빈도 를 적용한 경우에는 단어빈도, 로그 단어빈도의 사용에 관계없이 그리고 문서의 유사도를 산출하는 방식과 상관없이 문서검색보다 좋지 않은 성능을 보였다. 이 결과는 Wilkinson(1994)의 실험에서 역단락빈도를 적용하였을 때 문서검색보다 좋지 않은 성능을 보인 결과와도 일치한다.

노드검색 실험 결과를 제시한 〈표 2〉를 보면 로그 단어빈도 가중치를 적용하고 노드의 유사도 평균값을 문서의 유사도로 산출한 경우가 0.4669로 문서검색 결과 대비 14.66%의 성능 향상률을 나타내 가장 좋은 결과를 보여주었다. 노드검색에서 문서의 유사도 산출방식에 따른 검색성능 평가 결과, 최대값 0.4635, 평균값 0.4669, 그리고 혼합 0.4641로 거의 비슷한 성능을 보여 문서의 유사도 산출방식에 따른 검색결과의 차이는 그리 크지 않은 것으

〈표 2〉 노드검색의 실험 결과

			평균정확률	향상률	p(10)	p(20)
문서검색(ltf*idf)			0.4072	-	0.4710	0.3805
문서 유사도 산출 방식	최대값 (Max)	ltfe	0.4635	13.83%	**0.5284**	**0.4987**
		ltfe*ief	0.3621	-11.08%	0.4128	0.3897
	평균값 (Avg)	ltfe	**0.4669**	14.66%	0.5089	0.4870
		ltfe*ief	0.3417	-16.09%	0.3691	0.3677
	혼합 (Mix)	ltfe	0.4641	13.97%	0.5012	0.4850
		ltfe*ief	0.3573	-12.25%	0.3859	0.3752



〈그림 3〉 노드검색의 성능 비교

로 분석된다.

검색순위 상위 10위내에서는 노드의 유사도 최대값을 문서의 유사도로 산출했을 때 0.5284로 평균값 방식의 0.5089, 혼합 방식의 0.5012보다 상대적으로 좋은 성능을 보이는 것으로 나타났다. 이 결과는 검색결과 상위 20위내에서도 동일한 패턴을 보여주어 최대값을 문서의 유사도로 산출한 경우가 0.4987로 평균값 0.4870, 혼합 0.4850보다 좋은 성능을 나타냈다. 실제 정보검색시 이용자에게는 검색결과 상위에 나타나는 문서가 중요하므로 노드의 유사도 최대값을 문서의 유사도로 산출하는 방식이 유용할 것으로 보인다.

4.2 확장 노드검색 실험 결과

확장 노드검색 실험에서는 확장인자를 이용하여 하위 노드의 검색기회를 높이도록 하여 검색성능에 어떠한 차이를 보이는지 알아보았다. 노드검색 실험에서 노드의 유사도 최대값을 문서의 유사도로 간주할 경우 대부분 섹션 수준의 노드가 검색결과로 제시되었다. XML 검색이 충분히 의미가 있는 가장 작은 단위를 검색결과로 제시하려는 목적을 가지고 있다는 점에서 보면 노드검색은 XML 검색을 위한 좋은 방안이 되지는 못한다고 할 수 있다.

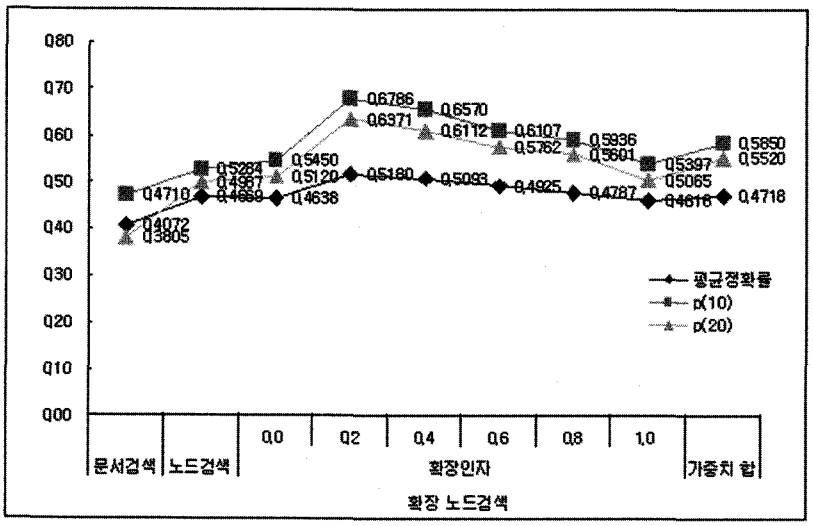
XML과 같이 여러 계층을 가지는 문서의 경우 중복색인을 방지하기 위해 독립색인 방식으로 색인을 작성하였을 때, 최하위의 문단에서 서브섹션과 섹션까지 연속적으로 가중치를 합하게 되면 결과적으로 상위의 노드가 검색결과로 제시될 확률이 높아지게 된다. 이러한 측면에서 여러 계층을 가지는 XML 문서에서는 하위 노드의 색인 가중치를 더하여 상위 노드의 색인 가중치로 부여하는 것보다 확장인자를 이용하여 색인 가중치를 계산하는 방법이 더 작은 단위의 노드를 검색결과로 갖는 기회를 높일 수 있게 된다.

확장인자 0.2에서 1.0까지 0.2씩 변화시켜가며 상위 계층에 출현하는 단어의 가중치를 재계산한 결과에 따르면 확장인자 0.2를 적용하였을 경우에 가장 좋은 성능을 보였다. 〈표 3〉과 〈그림 4〉에서 보듯이 확장인자 0.2를 적용하였을 경우 11-지점 평균정확률이 0.5180로 가장 좋은 성능을 보여 문서검색 대비 25.07%의 향상률을 나타냈다. 그 다음으로 확장인자 0.4를 적용하였을 경우가 0.5093로 문서검색 결과 보다 20.95%의 성능 향상률을 가져오는 좋은 결과를 보였다. 또한 확장인자 0.2와 0.4는 상위 10위에서의 정확률 p(10)에서 각각 0.6786과 0.6371을 상위 20위에서의 정확률 p(20)에서는 0.6570과 0.6112의 정확률을 보여 검색결과 상위 순위에서도 좋은 성능을 보임을 알 수 있다.

노드검색과 비교하였을 때에도 확장 노드검

〈표 3〉 확장 노드검색의 실험 결과

			평균정확률	향상률	p(10)	p(20)
문서검색			0.4072	-	0.4710	0.3805
노드검색			0.4669	14.66%	0.5284	0.4987
확장 노드검색	확장 인자	0.0	0.4638	13.90%	0.5450	0.5120
		0.2	**0.5180**	27.21%	**0.6786**	**0.6371**
		0.4	0.5093	25.07%	0.6570	0.6112
		0.6	0.4925	20.95%	0.6107	0.5762
		0.8	0.4787	17.56%	0.5936	0.5601
		1.0	0.4616	13.36%	0.5397	0.5065
가중치 합			0.4718	15.86%	0.5850	0.5520



〈그림 4〉 확장 노드검색의 성능 비교

색의 실험 결과는 확장인자 0.2~0.8을 적용한 경우에 문서검색 대비 17.56%~27.21%의 성능 향상률을 나타내어 노드검색 실험 결과의 성능 향상률 14.66%에 비해 좋은 결과를 보였다.

가중치 합, 즉 단순히 상하위 노드의 단어가중치를 더한 값을 상위 노드의 가중치로 부여하여 검색을 수행한 결과는 0.4718의 평균정확률을 나타냈다. 이 결과로 보았을 때 단순히 상하위 노드의 가중치 합을 상위 노드의 용어가중치로 부여하는 경우에는 하위 노드의 검색 기회를 제한할 뿐 아니라 검색성능에 있어서도 바람직하지 못함을 알 수 있다. 확장인자 0.0을 적용한 실험은 상하위 두 노드에 동시에 나타나는 용어에 대해 하위 노드의 용어가중치를 상위 노드의 용어가중치에 반영하지 않고 검색을 수행한 것으로 0.4638의 낮은 검색성능을 보여 주었다. 이는 하위 노드의 검색 기회는 높여주지만 상위 노드의 중요도를 제대로 반영하지 못하는 결과를 가져오게 되어 상대적으로 좋지 않은 검색성능을 보인 것으로 분석된다.

확장인자 1.0을 적용한 경우는 단순히 상위 색인 노드의 색인 가중치와 하위 색인 노드의 색인 가중치를 더한 것보다 상위 노드에 더 큰 가중치를 부여한다. 확장인자 1.0을 적용한 경우는 상하위 노드의 단순 가중치 합보다 더 큰 값을 상위노드의 용어가중치로 갖도록 하여 검색을 수행한 실험으로 검색 결과 0.4616으로 가장 좋지 않은 성능을 보여주었다. 이 결과로 알 수 있는 것은 상위 노드의 검색 기회를 상대적으로 높여 하위 노드의 검색 기회를 제한하는 경우가 가중치 합을 적용한 실험과 마찬가지로 검색성능에서도 좋은 성능을 가져올 수 없다는 것이다.

5. 결 론

이 연구에서는 구조문서를 구성하는 의미있는 정보단위인 노드를 검색대상으로 하고, 벡터공간모델 검색기법을 이용하여 실험을 수행하였다. 문서의 노드는 상하간의 중첩구조로 이루어지므로 실험집단의 색인방법 또한 문서검색의 색인방법과 달리 서브트리 색인과 독립색인의 두 가지 방식을 적용하였다.

단계별로 수행된 실험과 평가의 결과는 다음과 같다.

첫째, 실험문서의 색인을 작성하는 단계에서는 기초 실험 결과로 활용되는 문서검색을 위한 문서색인, 색인 서브트리 방식에 의한 노드 중복색인, 그리고 독립색인 방식에 의한 노드 독립색인 등 세 가지 색인을 수행하였다. 노드 독립색인은 구조 정보를 이용하는 확장 노드검색 실험에서 사용하였다.

둘째, 노드검색 실험에서는 문서 유사도 산출방식으로 노드 유사도의 최대값, 평균값, 그리고 노드 유사도와 문서 유사도의 혼합값 등 세 가지를 이용하였는데 서로 간에 성능차이는 거의 없는 것으로 나타났다. 다만 최대값을 적용했을 때 검색순위 상위에서 문서검색보다 좋은 성능을 보였다.

노드검색 실험에서는 적용한 가중치와 문서의 유사도 산출방식에 따른 검색 실험 결과 로그 단어빈도를 적용하고 노드 유사도 평균값을 문서의 유사도로 산출한 방법을 적용한 검색결과가 가장 좋은 성능을 보여 0.4669로 문서검색 결과 0.4072 대비 14.66%의 성능 향상률을 나타냈다. 노드는 문서보다 상대적으로 길이가 짧아 단락 내 단어빈도로 1이 많이 나타나게 되므로 단어빈도가 1인 용어의 낮은 영향력을 보완한 로그 단어빈도의 성능이 좋게 나타난 것으로 분석된다.

용어가중치 중 역노드빈도 가중치를 적용한 실험 결과는 노드검색에서 문서의 유사도를 산출하는 방식과 상관없이 초기 실험 결과보다 좋지 않은 성능을 보였다. 그 이유는 역문헌빈도 가중치를 적용하는 이유에서 찾을 수 있을 것이다. 문서들은 서로 다른 주제를 다루고 있어서 역문헌빈도 가중치를 사용하여 문서빈도가 낮은 단어, 즉 적은 수의 문서에 나타난 단어에 높은 중요도를 부여한다. 그러나 한 문서 내의 노드들은 유사한 개념을 표현하고 있으므로 역노드빈도를 적용하는 의미가 달라져 검색성능이 저하된 것으로 분석되었다. 이 결과는 Wilkinson(1994)의 연구결과와도 일치한다.

확장 노드검색 실험에서는 FERMI 모델에 기반하여 색인 단위를 정의하고 독립색인을 작성한 후 하위 노드의 검색기회를 높이기 위해 확장인자를 이용하여 상위 노드의 색인 가중치를 재계산하는 실험을 수행하였다. 작은 값의 확장인자를 적용했을 때 검색성능이 좋았는데, 특히 0.2를 적용했을 때 검색 평균정확률이 0.5210으로 문서검색 대비 27.21%의 성능 향상률을 나타냈다. 검색순위 상위에서도 높은 정확률을 보여 문서검색 대비 검색 순위 10위에서는 44.07%, 20위에서는 67.45%의 높은 성능 향상률을 나타냈다.

이상의 실험 결과를 통해서 이 연구에서는 노드 기반 XML 문서검색을 위한 가장 효율적인 검색 접근법으로 문서의 구조정보를 적용한 확장 검색을 제안한다. 확장인자를 적용하는 확장 노드검색은 검색 알고리즘의 간편성에 비

해 상당히 좋은 검색결과를 보여주어 XML 문서검색을 위한 가능성을 보여주었다. 실제 검색에서 유용성을 가지는 검색순위 상위의 결과에서도 구조 정보를 활용하여 노드검색을 수행하는 계층적 검색 알고리즘이 좋은 검색 성능을 보여주었다.

그러나 이 연구에서 제안한 검색 실험기법들은 문서의 논리적 구조가 비교적 분명한 논문 기사를 대상으로 하였으므로 INEX 컬렉션과 같은 대규모 실험집단에 적용하여 검증할 필요가 있다. 또한 실제 XML 문서가 사용되고 있는 웹 환경에서 이 실험에서 제안한 기법의 유용성을 검증할 필요가 있다.

또한, 확장 검색에 적용하기 위한 용어가중치에 대한 연구가 필요하다. 확장 노드검색 실험은 색인 용어 가중치가 내용기반 XML 검색을 위해 확장될 수 있는 방법을 보여주었다는 데 의의가 있다. 원칙적으로는 각 색인 노드마다 적용하는 확장인자가 달라야 하지만 각 색인 노드마다 특정 가중치를 적용할 것인가 하나의 전역 가중치를 적용할 것인가의 문제는 이론적인 수준이나 경험적 수준에서 수행될 문제이다. 확장인자를 유도하는 또 하나의 방법은 색인노드의 크기, 그리고 동일 수준 노드와 자식노드의 개수에 대한 정보에 기초하여 이루어질 수 있을 것이다. 실험에 사용하는 용어가중치를 정규화하는 방식에 따라 검색성능이 달라지므로 일반화할 수 있는 가중치 공식에 대한 연구가 필요할 것이다. 구조문서 검색을 위한 적합성 평가가 수행된다면 확장인자를 추정하기 위한 적합성 피드백 방법도 고려해 볼 수 있을 것이다.

참 고 문 헌

Abolhassani, M, and N, Fuhr. 2004. "Applying the Divergence from Randomness Approach for Content-Only Search in XML Documents." *26th European Conference on Information Retrieval Research (ECIR 2004).* Springer.

Billingsley, P. 1979. *Probability and Measure*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc, New York.

Callan, James P. 1994. "Passage-Level Evidence in Document Retrieval." *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Carmel, D., Y. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. 2003.

"Searching XML documents via XML fragments." *In Proceedings of the 26th ACM SIGIR Conference,* 151-158.

Chiaramella, Y., P. Mulhem, and F. Fourel. 1996. *A Model for multimedia information retrieval.* Technical report, FERMI ESPRIT BRA 8314, University of Glasgow.

Fuhr, N., N. Gövert, N. Rölleke, T. 1998. "DOLORES: A System for Logic-Based Retrieval of Multimedia Objects." *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 257-265.

Hatano, K., H., M. Kinutani, M. Watanabe Yoshikawa, and S. Uemura. 2002. "Determining the Unit of Retrieval Results for XML Documents." *In Proceedings of the First INitiative for the Evaluation of XML Retrieval (INEX).*

Huang, F., S. Watt, D. Harper, and M. Clark. 2006. "Robert Gordon University at INEX 2006: Adhoc Track." *In Proceedings of the First INitiative for the Evaluation of XML Retrieval (INEX).*

INEX(Initiative for the Evaluation of XML retrieval). 〈http://inex.is.informatik.uni-duisburg.de/〉

INEX(Initiative for the Evaluation of XML retrieval). 2006. 〈http://inex.is.informatik.uni-duisburg.de/2006/〉

Kamps, J., M. de Rijke, and B. Sigurbjörnsson. 2004. "Length normalization in XML retrieval." In *Proceedings of the 27th Annual International ACM SIGIR Conference.*

Kamps, J., M. Marx, M. de Rijke, and B. Sigurbjorolsson. 2003. "XML Retrieval: What to Retrieve?" *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval.*

Kaszkiel, M. and J. Zobel. 1997. "Passage retrieval revisited." *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 178-185.

Kazai, G., M. Lalmas, and T. Rolleke. 2002. "Focused Structured Document Retrieval." In *Proceedings of the 9th*

Symposium on String Processing and Information Retrieval(SPIRE 2002), Springer, 241-247.

Korfhage, R. 1997. *Information storage and retrieval.* NY : Wiley.

Luk, Robert W.P., H. V. Leong, Tharam S. Dillon, T. S. Alvin, W. Chan, Croft, Bruce, and Allan James. 2002. "A survey in indexing and searching XML documents." *Journal of the American Society for Information Science and Technology,* 53(6): 415-437.

Mass, Y., M. E. Mandelbrod, D. Amitay, Y. Maarek Carmel, and A. Soffer. 2002. "JuruXML - an XML retrieval system at INEX'02." *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval.*

Mass, Y. and M. Mandelbrod. 2003. "Retrieving the most relevant XML Components." *Proceedings of the third Workshop of the INitiative for the Evaluation of XML Retrieval.*

Salton, G, J. Allan, and C. Buckley. 1993. "Approach to passage retrieval in full text information systems."Proceedings of the 16th Annual International *Conference on Research and Development in Information Retrieval.*

Wilkinson, R. 1994. "Effective retrieval of structured documents." *Proceedings of SIGIR Conference,* 311-317.

Wolff, J.E., H. Florke, and A. B. Cremers. 2000. "Searching and browsing collections of structural information." *Proceedings of IEEE advances in digital libraries.*