XML 데이터의 2단계 인덱싱 기법

이범석, 황병연
가톨릭대학교 컴퓨터공학과
e-mail:{bslee,byhwang}@catholic.ac.kr

Two-step Indexing Method for XML data

Bum-Suk Lee, Byung-Yeon Hwang
Dept. of Computer Engineering, The Catholic University of Korea

요 약

XML은 웹2.0 환경에서 데이터의 저장과 전달을 위한 역할을 수행하는 필수적인 포맷으로 각광받고 있다. 특히 RSS나 ATOM과 같은 피드기술은 XML을 이용한 성공적인 사례로 인정받고 있다. 이러한 XML 포맷 데이터는 빠른 검색을 위해 경로기반 클러스터링 기법이나 내용기반 클러스터링 기법을 적용하는 것이 일반적이다. 하지만 클러스터링 기법을 적용할 때 주어지는 임계값에 따라 재현율이 변화하게 되고, 검색 결과에서 배제되는 데이터가 발생하게 된다. 이 논문에서는 기존 클러스터링 기법을 적용할 때 발생하는 데이터 배제현상을 보완하는 2단계 인덱싱 기법을 제안하고, 제안한 방법의 성능에 대해 분석한다.

1. 서론

XML 표준은 웹2.0 환경의 발달과 함께 그 사용도 증가하였는데, 특히 데이터의 전달과 저장이라는 기본적인 기능을 수행할 뿐만 아니라 RSS나 ATOM같은 피드의 기반 기술로 사용되면서 웹 서비스 개발에서 필수적인 역할로 자리매김 하였다.

지난 몇 년동안 XML 데이터의 저장에 관한 연구는 반구조적인 데이터를 관계형 DBMS에 효율적으로 매핑하는 것에 중점을 두고 진행되었다. 이러한 연구들에는 빠른 비트 연산을 이용한 비트맵인덱스[1,2]와 문서 검색에 좋은 성능을 가진 역인덱스[3,4], 그리고 XML 데이터의 구조적 특징을 반영한 그래프 인덱스[5,6,7] 등이 있다. 특히 3차원 비트맵인덱스 기법을 적용한 BitCube[1]의 경우 빠른 검색속도 뿐만 아니라 XML 문서의 구조적 유사성을 기반으로 클러스터링을 수행하여 XPath 질의에 효율적으로 대응하는 결과를 보여주었다. 이처럼 다양한 구조를 가지는 XML 데이터의 처리는 이질적인 데이터의 교환이나, 웹서비스 통합 등의 분야에서 좋은 결과를 가져올 수 있다. 하지만 최근 사용되는 XML 데이터는 XML을 기반으로 표준화된 스키마가 존재하는 경우가 대부분이다. 이러한 경우 XQuery 질의는 기본적인 경로 검색 기능으로 사용될 뿐이고, 검색 속도를 좌우하는 것은 그 경로에 존재하는 내용을 얼마나 잘 인덱싱하느냐에 따라 결정된다.

본 논문에서는 XML 문서의 내용을 효율적으로 클러스터링 하기 위한 2단계 인덱싱 기법을 제안한다. 첫 번째 단계의 인덱싱은 문서의 고유번호와 해당 내용에 포함된 단어로 비트맵인덱스를 생성하고 클러스터링을 수행한다. 두 번째 단계에서는 이전 단계에서 서로 다른 클러스터에 저장된 데이터에 대해 단어를 기준으로 인덱스 리스트를 구성한다. 이 두 단계의 인덱싱 방법은 빠른 비트 연산을 이용하면서도 클러스터링 때문에 검색 결과에서 배제되는 데이터도 모두 단어 인덱스 리스트를 이용하여 검색이 가능해진다.

본 논문에서는 제안하는 2단계 인덱싱 기법에 대해 자세히 소개하고, 기존의 방법과 제안한 방법에 어떠한 차이가 있는지 성능을 비교한다. 논문의 구성은 다음과 같다. 2장에서는 관련연구를 소개하고, 3장에서는 2단계 인덱싱 기법을 제안한다. 4장에서는 제안한 방법의 간단한 성능분석 및 결론과 향후 연구 계획에 대해 제시한다.

2. 관련연구

XML에서 클러스터링에 관한 연구는 XPath 검색의 효율을 높이기 위해 경로의 유사도에 기반한 클러스터링 기법[8]과 내용 검색의 효율을 높이기 위한 내용 기반 클러스터링 기법[9]으로 나누어진다. 그 중 문서, 경로, 내용을 축으로 하는 3차원 비트맵인덱스 기법은 비트 연산을 이용한 빠른 검색 성능을 가지는 대표적인 경로 기반 인덱싱 기법이지만, 3차원으로 구성되기 때문에 저장되는 XML 문서가 많아지면 메모리의 점유가 급격히 증가한다. 경로 비트맵인덱싱[10]은 이러한 문제점을 해결하기 위해 포인터를 이용하여 3차원 비트맵인덱스를 개선하였다.

내용 클러스터링 기법은 주어진 임계값을 기준으로 데이터를 여러 클러스터로 분할하고 그만큼의 검색 속도 향상을 기대할 수 있다. 하지만 임의로 주어진 임계값이나 클러스터의 수만큼으로 데이터를 클러스터링 하기 때문에, 검색의 재현율이 가변적이라는 문제점을 가진다. 재현율의

정의는 다음과 같다.

정의 1. *재현율(recall)은 검색을 위해 입력한 키워드와 관련된 전체 데이터 중에서 검색 결과에 포함된 데이터의 비율을 의미한다.*

recall = (retrieved data ∩ related data) / related data

3. 2단계 인덱싱

이 논문에서는 클러스터링된 데이터의 검색 재현율을 향상시키기 위해 2단계 인덱싱 구조를 제안한다. 첫 번째 인덱싱 단계에서는 문서와 단어를 이용하여 비트맵인덱스를 구성하여 문서에 포함된 단어의 유사도를 기준으로 클러스터링을 수행한다. 두 번째 단계에서는 각 단어에 대해 클러스터 대푯값에서 제외된 것을 검색하여 단어 인덱스 리스트로 구성한다.

(그림 1)은 이 논문에서 설명을 위해 사용할 XML 문서의 예와 그것을 관계형 DBMS의 테이블에 파싱하여 저장한 모습을 보여준다.

d_1	<element id="1">t_1 t_3 t_4 t_6 t_8 t_{10}</element>
d_2	<element id="2">t_2 t_3 t_4 t_6 t_8 t_9</element>
d_3	<element id="3">t_1 t_3 t_4 t_5 t_7 t_8 t_{10}</element>
d_4	<element id="4">t_1 t_3 t_4 t_5 t_{10}</element>
d_5	<element id="5">t_2 t_4 t_5 t_6 t_9</element>
d_6	<element id="6">t_2 t_4 t_5 t_6 t_7 t_8 t_9</element>

⇓

docID	*elementID*	*content*
d_1	1	t_1 t_3 t_4 t_6 t_8 t_{10}
d_2	2	t_2 t_3 t_4 t_6 t_8 t_9
	...	
d_6	6	t_2 t_4 t_5 t_6 t_7 t_8 t_9

(그림 1) XML 데이터와 관계형 DBMS 테이블

3.1 1단계; 비트맵인덱스 클러스터링

첫 번째 단계에서는 문서의 고유번호와 문서에 포함된 단어를 이용한 비트맵인덱스를 생성하고, 내용의 유사도를 기준으로 클러스터링을 수행한다. (그림 2)는 (그림 1)에서 제시한 6개의 XML 문서와 각 문서(d_1~d_6)를 구성하는 10개의 단어들(t_1~t_{10})을 이용하여 구성한 비트맵인덱스와 클러스터링을 수행한 후 생성된 두 개의 클러스터(C_1,C_2)를 보여준다.

비트맵인덱스를 구성할 때 단어는 포함된 모든 단어이거나, 자동화된 태그 추출을 이용해서 선별한 단어가 될 수 있다. 비트맵인덱스의 클러스터링을 위한 유사도는 $1-|xOR(d_i,d_j)/MAX([d_i],[d_j])|$로 계산된다. 예를 들어 d_1과 d_2의 유사도는 1-4/10=0.6이고, d_1과 d_3의 유사도는 1-3/10=0.7이다. (그림 2)에서 클러스터링을 수행하기 위한 유사도 임계값을 0.67이상으로 설정하였다. 클러스터링 후의 R_1과 R_2는 각 클러스터의 비트 대푯값을 의미하며, 비트 대푯값을 산출하기 위한 임계값은 0.5이상이다.

【클러스터링 수행 전】

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
d_1	1	0	1	1	0	1	0	1	0	1
d_2	0	1	1	1	0	1	0	1	1	0
d_3	1	0	1	1	1	0	1	1	0	1
d_4	1	0	1	1	1	0	0	0	0	1
d_5	0	1	0	1	1	1	0	0	1	0
d_6	0	1	0	1	1	1	1	1	1	0

⇓ 【클러스터링 수행 후】

	t_1	t_3	t_4	t_5	t_6	t_7	t_8	t_{10}	
d_1	1	1	1	0	1	0	1	1	
d_3	1	1	1	1	0	1	1	1	C_1
d_4	1	1	1	1	0	0	0	1	
R_1	1	1	1	1	0	0	1	1	

	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	
d_2	1	1	1	0	1	0	1	1	
d_5	1	0	1	1	1	0	0	1	C_2
d_6	1	0	1	1	1	1	1	1	
R_2	1	0	1	1	1	0	1	1	

(그림 2) 비트맵인덱스 클러스터링

이 상태에서 t_5를 포함한 문서를 검색하기 위한 질의가 입력되면 먼저 대푯값을 확인하여 두 개의 클러스터를 모두 검색하여 결과를 반환한다. 이때는 재현율이 낮아지는 문제가 발생하지 않는다. 재현율이 낮아지는 예로 t_3을 검색한다면 대푯값에 t_3이 존재하는 클러스터 C_1의 데이터를 검색해서 d_1, d_3, d_4를 반환한다. 클러스터링을 수행하기 전에 비해 연산시간은 줄어들 수 있지만, 클러스터링 전에 1이었던 재현율은 클러스터링 후에 0.75로 낮아지는 문제점이 발생한다. 또한 t_7을 검색하면 재현율은 0이 된다.

3.2 2단계; 단어 인덱스 리스트 생성

두 번째 단계에서는 재현율이 낮아지는 문제점을 보완하기 위해서, 클러스터링을 수행한 후에 각 단어에 대해 인덱스 리스트를 구성한다. 단어 인덱스 리스트는 비트 대푯값에서 제외된 단어를 기준으로 생성하는데, (그림 2)의 예에서는 t_3, t_6, t_7의 세 단어가 그 대상이 된다. (그림 3)은 단어 인덱스 리스트의 구조를 보여준다.

t_3	→	d_2		...	
t_6	→	d_1		...	
t_7	→	d_3	d_6	...	

(그림 3) 단어 인덱스 리스트

4. 결론 및 향후연구계획

본 논문에서 제안한 2단계 인덱싱 방법은 검색에 대한 재현율을 항상 1이 될 수 있도록 비트맵인덱스와 단어 인덱스 리스트를 함께 유지하고, 질의에 대해 두 인덱스를 함께 검색하여 모든 데이터를 반환한다.

클러스터링을 수행한 후의 검색시간은 평균적으로 전체 데이터를 검색하는 시간을 클러스터의 수로 나눈 만큼이 소요된다. 2단계 인덱싱에서는 단어 인덱스 리스트를 검색하는 시간이 추가로 소요될 수 있지만, 단어 인덱스를 단순한 리스트로 구현하더라도 $O(n)$의 선형시간만큼 소요되므로 큰 오버헤드가 발생하지는 않는다.

향후 연구로는 실제 데이터에 적용한 실험이 진행 중이며, 효율적인 클러스터링을 위한 적절한 임계값을 설정하는 것도 중요한 연구과제이다.

참고문헌

[1] J. Yoon, V. Raghavan, V. Chakilam, and L. Kerschberg, “BitCube: A Three-Dimensional Bitmap Indexing for XML Documents,” Journal of Intelligent Information System, Vol.17, pp. 241-254, 2001.

[2] J. Yoon, V. Raghavan, and V. Chakilam, “BitCube: Clustering and Statistical Analysis for XML Documents,” In Proc. of the 13th International Conference on Scientific and Statistical Database Management, Fairfax, Virginia, July 2001.

[3] 민경섭, 김형주, “상이한 구조의 XML 문서들에서 경로 질의 처리를 위한 RDBMS 기반 역인덱스 기법,” 정보과학회논문지, 제30권 제4호, pp. 420-428, 2003.

[4] 서치영, 이상원, 김형주, “XML 문서에 대한 RDBMS에 기반을 둔 효율적인 역색인 기법,” 정보과학회논문지, 제30권 제1호, pp. 27-40, 2003.

[5] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom, “Lore: A Database Management System for Semistructured Data,” ACM SIGMOD Record, Vol.26, No.3, pp. 54-66, 1997.

[6] C. Chung, J. Min, and K. Shim, “APEX: An Adaptive Path Index for XML Data,” In Proc. of the International Conference on ACM SIGMOD, pp. 121-132, Madison, Wisconsin, June 2002.

[7] R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes, “Exploiting Local Similarity for Indexing Paths in Graph-Structured Data,” In Proc. of the 18th IEEE International Conference on Data Engineering, pp. 129-140, 2002.

[8] T. Dalamagas, T. Cheng, K. J. Winkel, and T. Sellis, “A Methodology for Clustering XML Documents by Structure,” Information Systems, Vol.31, Issue 3, Elsevier Science Ltd., pp. 187-228, May 2006.

[9] T. Tran, R. Nayak, and P. Bruza, “Combining Structure and Content Similarities for XML Document Clustering,” In Proc. of the 7th Australasian Data Mining Conference, pp. 219-226, 2008.

[10] J. Lee and B. Hwang, “Path Bitmap Indexing for Retrieval of XML Documents,” Lecture Notes in Computer Science, Vol.3885, Springer-Verlag, April 2006.