www.calsec.or.kr
http://dx.doi.org/10.7838/jsebs.2012.17.3.061

시그니쳐를 이용한 XML 경로 비교의 최소화 기법

A Minimization Technique of XML Path Comparison Based on Signature

장경훈(Kyunghoon Jang)*, 황병연(Byung-Yeon Hwang)**

초 록

XML은 사용자가 태그를 자유롭게 정의할 수 있어 다양한 구조의 문서가 생성된다. 이렇게 작성된 XML 문서를 효율적으로 관리하기 위해 경로의 유사도에 기반한 클러스터링 및 검색에 대한 연구가 진행되어 왔다. 이에 대한 초기 연구인 3차원 비트맵 인덱싱 기법에서는 유사한 XML 문서를 클러스터링하고 검색하기 위해 경로가 인덱스를 구성하는 단위로 사용되었다. 이 기법은 XML 문서의 구조가 변경되었을 때 변경 전의 경로와 변경 후의 경로가 정도에 상관없이 완전히 다른 것으로 인식되는 문제점이 있다. 이에 따라 경로의 유사도를 측정하는 기법들이 제안되었다.

경로의 유사도를 측정하기 위해서는 비교 대상이 되는 두 경로의 노드들을 비교하는데, 이 과정에서 두 경로에 공통으로 존재하지 않는 노드로 인해 불필요한 비교가 발생한다. 하지만 기존의 경로 유사도 기법들은 이러한 점을 고려하지 않았다. 이를 해결하기 위해 본 논문에서는 시그니쳐를 이용하여 경로 간 노드의 비교를 최소화하는 기법을 제안한다. 실제 다양한 구조의 XML 문서를 가지고 성능 평가를 실시하여, 본 논문에서 제안한 기법을 이용했을 때 기존 기법을 이용했을 때보다 약 20%의 성능을 개선시켜 제안한 기법의 우수함을 보인다.

ABSTRACT

Since XML allows users to define any tags, XML documents with various structures have been created. Accordingly, many studies on clustering and searching the XML documents based on the similarity of paths have been done in order to manage the documents efficiently. To retrieve XML documents having similar structures, the three-dimensional bitmap indexing technique uses a path as a unit when it creates an index. If a path structure is changed, the technique recognizes it as a new path. Thus, another technique to measure the similarity of paths was proposed.

To compute the similarity between two paths, the technique compares every node of the paths. It causes unnecessary comparison of the nodes, which do not exist in common between the two paths. In this paper, we propose a new technique that minimizes the comparison using signatures and show the performance evaluation results of the technique. The comparison speed of proposed technique was 20 percent faster than the existing technique.

키워드 : XML, 시그니쳐, 유사 경로, 클러스터링, 노드 비교, 성능평가
XML, Signature, Similar Path, Clustering, Node Comparison, Performance Evaluation

---

본 연구는 2012년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

* NHN I&S 사내정보시스템센터
** 교신저자, 가톨릭대학교 컴퓨터정보공학부 교수

2012년 06월 12일 접수, 2012년 08월 06일 심사완료 후 2012년 08월 10일 게재확정.

1. 서 론

1998년에 W3C에서 표준으로 제정된 XML (Extensible Markup Language)[7]은 확장 가능한 마크업 언어로 인터넷의 대중화에 기여한 HTML(Hyper-Text Markup Language)이 발전한 형태라 할 수 있다. HTML은 단순하고 쉬워서 널리 사용되었지만 특수 문자나 기호 등 복잡한 기능을 수행하지 못하는 단점이 있다. 이를 보완하기 위해 사용자 임의로 태그를 정의하여 문서의 구조를 정의할 수 있는 SGML(Standard Generalized Markup Language)이 개발되었으나 복잡하고 활용하기 어려운 단점이 있었다. 이를 해결하기 위해 개발된 것이 XML이다. XML은 자기 서술적인(self-describing) 특징으로 문서의 내용을 다양한 형태로 보여줄 수 있을 뿐만 아니라, 플랫폼과 프로그램으로부터 독립적이며 개방 표준이고, 내용, 구조, 형태의 분리 개념에 실제적으로 접근할 수 있는 장점이 있다. 이에 따라 XML은 전자상거래, 인터넷 뱅킹, 통신 규약, 정보 교환 및 전자 서적 등 다양한 분야에서 사용되고 있다. 그러나 자유롭게 구조를 만들고 변경할 수 있는 XML의 유연성은 다양한 구조를 갖는 XML 문서의 확산을 초래하였고, 이를 효율적으로 저장 및 검색하기 위한 시스템의 연구가 진행되어 왔다 [1, 3, 5, 8, 10, 11].

이와 관련된 연구로 3차원 비트맵 인덱스를 이용한 BitCube[12]는 XML 문서에 대한 문서 d, 경로 p, 단어 w를 기준으로 (d, p, w) 쌍의 값이 존재하면 1, 존재하지 않으면 0의 값을 갖는 3차원 비트맵 인덱스를 생성한다. 그래서 문서가 포함하고 있는 경로 p의 존재 유무에 따라 유사도를 계산하고 유사한 경로를 포함한 문서들끼리 클러스터링을 수행한다. 이 시스템은 다른 상용 XML 문서 검색 시스템들과의 성능 평가에서 우수한 성능을 입증하였다. 그러나 BitCube는 문서의 경로에 새로운 노드가 추가되었을 때 그 구조가 유사하지만 서로 다른 구조로 인식하게 되는 문제점을 가지고 있다. 이러한 문제점을 해결하고자 XML 문서의 유사한 구조를 갖는 경로의 클러스터링 및 검색을 효율적으로 하기 위해 경로의 유사도를 측정하는 기법[2, 4, 9]들이 제안되었다. Lee and Hwang[4]에서는 비교되는 두 경로의 모든 노드에 대해 편집 거리를 구하여 유사도를 측정하였고, Kim et al.[2]에서는 상대적 위치가 같은 경로 간의 공통 노드들의 점수(Position)와 상대적 위치가 다른 경로 간의 공통 노드들의 점수(Node)를 고려하여 경로 간 의미적 유사도를 측정하였다.

경로 유사도를 측정하기 위해서는 두 경로 간 노드를 비교해야 한다. XML 문서는 작성자의 의도에 따라 다양한 구조와 용어가 사용되기 때문에 서로 다른 문서의 경로 간 노드를 비교하면 노드 간 서로 일치하지 않는 경우가 더 많게 된다. 기존 유사도 측정 기법들에서는 이러한 특성을 고려하지 않고 모든 노드를 일일이 비교하기 때문에 성능이 저하된다. 본 논문은 이러한 문제점을 해결하고자 시그니처[6]를 이용한 경로 간 노드의 비교 수를 최소화하는 기법을 제안하고, Kim et al.[2]에 제안 기법을 적용했을 때와 적용하지 않았을 때의 성능을 비교 분석한 결과 약 20%의 성능 개선이 있음을 입증하였다.

본 논문의 제 2장에서는 관련 연구인 XML 문서의 클러스터링 및 검색과 시그니처 기법

을 살펴본다. 제 3장에서는 제안하는 기법으로 경로 시그니쳐를 구성하는 방법에 대해 설명하고, 이를 경로 비교에 적용하는 예를 보인다. 또한, 제안 기법에서 발생하는 문제점과 이에 대한 해결 방안을 제시한다. 제 4장에서는 실제 데이터를 가지고 실험을 실시하여 제안한 기법의 성능이 우수함을 보인다. 마지막으로 제 5장에서는 결론에 대해서 언급하기로 한다.

2. 시그니쳐 기법

이 장에서는 본 논문에서 제안하는 기법에 이용되는 시그니쳐 기법에 대해 논의한다.

시그니쳐 기법은 정보 추출 및 데이터베이스 분야에서 많이 연구되어 왔다. 이 기법의 기본적인 아이디어는 개략적 필터(inexact filter)에 기반을 두고 있다. 개략적 필터 방식이란 질의 결과에 자격이 되지 않는 데이터를 우선 배제하여 검색에서 제외하겠다는 것이다.

시그니쳐 기법의 대표적인 방법에는 Superimposed Coding 방법과 Word Signature 방법이 있는데 본 논문에서는 Superimposed Coding 방법을 이용하였다. Superimposed Coding 방법은 <표 1>과 같다. 이 방법은 bit-wise OR 연산을 통해 모든 키워드에 대한 시그니쳐를 중첩시켜 블록 시그니쳐를 생성한다.

<표 1> Superimposed Coding 방법

Keyword	Signature
database	00110010
xml	10000101
twitter	10000110
Block signature	10110111

<표 1>은 문서에 있는 각 키워드에 대해서 해시 함수로 시그니쳐를 생성하고, 식 (1)과 같이 키워드 database, xml, twitter의 시그니쳐 값에 대해 bit-wise OR 연산을 수행하여 하나의 블록 시그니쳐를 구성한 것을 나타낸 것이다.

$$00110010 \vee 10000101 \vee 10000110 = 10110111 \quad (1)$$

문서에 특정 키워드가 포함되었는지에 대한 질의를 하는 경우, 질의에 사용된 키워드에 대한 시그니쳐를 생성하고 블록 시그니쳐와 bit-wise AND 연산을 수행한다. 그 결과가 질의문에 사용된 키워드의 시그니쳐와 일치한다면 문서가 그 키워드를 포함한다는 것을 알 수 있다.

3. 시그니쳐를 이용한 XML 경로 비교의 최소화 기법

이 장에서는 경로 시그니쳐를 구성하여 이를 경로 간 비교에 적용하는 방법과 시그니쳐 기법으로 인해 발생하는 문제점에 대해 논의한다. 본 논문에서 제안하는 기법은 1) 경로 시그니쳐를 구성하는 단계와 2) 경로 시그니쳐를 이용한 경로 비교 단계로 나누어진다. 경로 시그니쳐를 구성하는 단계는 데이터베이스에 저장될 XML 문서에서 추출된 경로와 클러스터의 대표 경로 간 비교하기 전에 처리되는 단계이고, 경로 시그니쳐를 이용한 비교 단계는 두 경로 간 노드를 비교하는 과정

에서 처리되는 단계이다.

3.1 경로 시그니쳐 구성

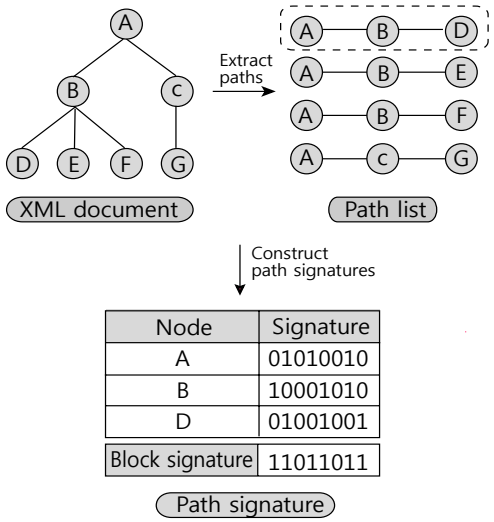
첫 번째 단계에 구성되는 경로 시그니쳐는 <표 2>와 같이 기준 경로 시그니쳐(Criteria path signature)와 대표 경로 시그니쳐(Representative path signature)로 구분된다. 기준 경로는 DOM Parser를 통해 데이터베이스에 저장할 XML 문서에서 추출한 경로를 말한다. 그리고 대표 경로는 유사한 경로들이 저장된 각 클러스터를 대표하는 경로를 말한다.

〈표 2〉 기준 경로와 대표 경로

기준 경로	데이터베이스에 저장될 XML 문서에서 추출된 경로
대표 경로	클러스터를 대표하는 경로

기준 경로 시그니쳐를 구성하는 과정은 다음과 같다. 먼저 데이터베이스에 저장할 XML 문서의 경로들을 추출하여 경로 리스트(Path list)를 얻고, 경로 리스트에 있는 경로들의 노드로부터 시그니쳐를 구한다. 그리고 각 노드의 시그니쳐 간 bit-wise OR 연산을 수행하여 블록 시그니쳐를 생성한다. 이 블록 시그니쳐는 기준 경로를 대표하는 시그니쳐로써 대표 경로 시그니쳐와 비교할 때 필터 역할을 하게 된다.

<그림 1>은 경로 리스트 중 하나의 경로 시그니쳐를 구성하는 과정을 나타낸다. 처음 XML 문서에서 4개의 경로가 추출되고 이를 포함한 경로 리스트가 만들어진다. 그리고 첫 번째 경로인 A/B/D에 대한 경로 시그니쳐는 다음과 같이 구성된다. 먼저 해시 함수를 통



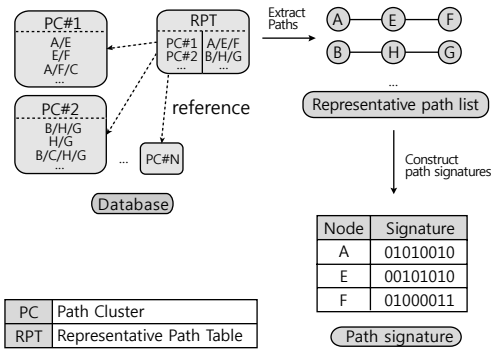
〈그림 1〉 기준 경로 시그니쳐 구성

해 각 노드별 시그니쳐(01010010, 10001010, 01001001)를 구하고 이 시그니쳐를 bit-wise OR 연산하여 블록 시그니쳐를 생성한다. 그리고 경로 리스트에 있는 나머지 경로 A/B/E, A/B/F, A/C/G에 대해서도 동일한 방법으로 경로 시그니쳐를 구성한다.

기준 경로 시그니쳐를 구성하면 기준 경로 시그니쳐와 비교할 대표 경로 시그니쳐를 구성해야 한다. 대표 경로 시그니쳐를 구성하려면 먼저 각 클러스터로부터 대표 경로를 추출해야 하는데, 본 논문에서는 Lee and Hwang [9]에서 제안한 대표 경로 추출 기법을 사용하였다. 대표 경로가 추출되면 기준 경로 시그니쳐를 구성할 때와 마찬가지로 경로를 구성하는 노드별로 시그니쳐를 구하고, 블록 시그니쳐는 생성하지 않는다. 그리고 대표 경로 시그니쳐는 데이터베이스에 존재하는 클러스터의 수만큼 구성된다.

<그림 2>는 N개의 대표 경로 중 하나의 대표 경로 시그니쳐를 구성하는 과정을 보여

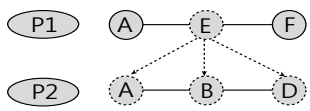
준다. 데이터베이스에는 N개의 경로 클러스터(Path cluster)가 있고, 각 경로 클러스터에는 다양한 XML 문서들로부터 추출된 유사한 경로들이 저장되어 있다. 그리고 대표 경로 테이블(Representative path table)에는 N개의 경로 클러스터의 대표 경로들이 저장되어 있다. 경로 시그니쳐의 구성은 먼저 대표 경로 테이블에서 대표 경로를 추출하여 N개의 경로를 포함한 대표 경로 리스트를 만든다. 그리고 첫 번째 대표 경로인 A/E/F의 각 노드에 대해 시그니쳐를 구하면 하나의 대표 경로 시그니쳐가 완성 된다. 나머지 대표 경로도 동일한 방법으로 수행한다.



〈그림 2〉 대표 경로 시그니쳐 구성

3.2 경로 시그니쳐를 이용한 경로 비교

이 절에서는 기존의 방법에서 발생되는 문제점을 논의하고 제 3.1절에서 구성한 기준 경로 시그니쳐와 대표 경로 시그니쳐를 경로 비교에 적용하는 방법을 소개한다. XML 문서는 사용자가 임의로 태그를 정의할 수 있기 때문에 다양한 구조의 문서가 만들어진다. 이러한 특징 때문에 서로 다른 XML 문서로부터 추출된 경로 사이에 공통으로 존재하지



〈그림 3〉 불필요한 노드 비교의 발생

않는 노드가 존재하게 된다.

예를 들어 <그림 3>의 경로 P1(A/E/F)과 P2(A/B/D)를 비교할 때 경로 P1의 노드 E는 경로 P2에 존재하지 않지만 비교하기 전에는 이 사실을 알 수 없다. 그래서 경로 P2의 노드 A, B, D와 모두 비교하고 나서야 일치하는 노드가 없다는 것을 알 수 있다. 클러스터가 100개가 있다면 기준 경로는 100개의 대표 경로와 비교해야 하는데, 그만큼 불필요한 노드의 비교가 발생할 것이다. 이러한 문제를 해결하기 위해 제 3.1절에서 구성한 기준 경로 시그니쳐와 대표 경로 시그니쳐를 사용한다. 대표 경로의 각 노드가 기준 경로에 일치하는 노드가 존재하는지 여부는 기준 경로 시그니쳐의 블록 시그니쳐와 대표 경로 시그니쳐의 각 노드 시그니쳐 간 bit-wise AND 연산을 통해 쉽게 확인할 수 있다.

기준 경로

Node	Signature
A	01010010
B	10001010
D	01001001
Block signature	11011011

대표 경로

Node	Signature
A	01010010
E	00101010
F	01000011

〈그림 4〉 기준 경로 시그니쳐와 대표 경로 시그니쳐

<그림 4>는 제 3.1절에서 구성한 두 경로 시그니쳐이다. 두 경로 시그니쳐를 이용한 경로 간 비교는 다음과 같이 행해진다. 먼저 식 (2)와 같이 대표 경로의 노드 A를 기준 경로

의 블록 시그니쳐와 bit-wise AND 연산을 수행하여 대표 경로의 노드 A가 기준 경로에 존재하는지의 여부를 확인한다. 그리고 그 결과가 노드 A 시그니쳐와 동일하기 때문에 노드 A가 기준 경로에도 존재한다는 것을 알 수 있다.

$$11011011 \wedge 01010010 = 01010010 \qquad (2)$$

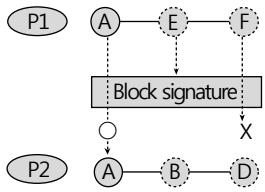
다음으로 대표 경로의 노드 E 시그니쳐를 식 (3)과 같이 기준 경로의 블록 시그니쳐와 bit-wise AND 연산을 수행하면 그 결과가 노드 E 시그니쳐와 다르게 나온다. 이는 기준 경로의 블록 시그니쳐가 노드 A, B, D만으로 구성되었기 때문이다. 이 결과를 통해 대표 경로의 노드 E가 기준 경로에 존재하지 않는다는 것을 확인하여 대표 경로의 노드 E와 기준 경로에 있는 노드 A, B, D와 비교하지 않고 다음을 진행할 수 있다.

$$11011011 \wedge 00101010 \neq 00101010 \qquad (3)$$

이와 같이 제안 기법을 사용하지 않고 경로간 비교하였을 때 특정 노드가 두 경로에 존재하는지의 여부와 상관없이 모든 노드를 비교한 후에야 존재 여부를 알 수 있었지만, 제안 기법을 이용하여 먼저 특정 노드의 존재 여부를 확인함으로써 불필요한 비교 수행을 줄였다.

3.3 제안 기법의 문제점과 이중 블록 시그니쳐

블록 시그니쳐를 생성할 때 경로를 구성하는 여러 개의 노드 시그니쳐들이 bit-wise OR 연산으로 취합되기 때문에 블록 시그니쳐 생성에 포함되지 않았던 노드의 시그니쳐와 bit-wise AND 연산의 결과가 우연히 매치되는 현상이 발생할 수 있다. 이것을 허위 통과(false drop) 또는 허위 적중(false hit)이라고 한다. 제안 기법에서 실제로 허위 통과가 발생하면, 노드 비교 외에 경로 시그니쳐 비교만큼 비교 시간이 늘어나는 단점이 있다. 제안 기법을 사용하지 않았을 때보다 비교 횟수가 늘어나게 된다. 그래서 이 절에서는 허위 통과의 발생을 줄이고자 이중 블록 시그니쳐에 대해 논의한다.



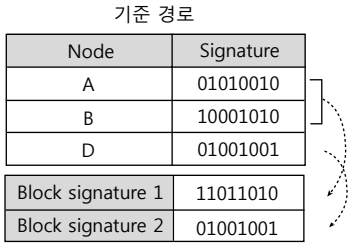
〈그림 5〉 노드 F의 허위 통과 발생

<그림 4>에서 대표 경로의 노드 F가 기준 경로에 존재하는지 확인하기 위해 식 (4)와 같이 기준 경로의 블록 시그니쳐와 bit-wise AND 연산을 수행했을 때, 결과가 F의 시그니쳐와 일치하는 것으로 나타나지만, 실제로 기준 경로에는 노드 F가 존재하지 않다. <그림 5>는 식 (4)처럼 허위 통과가 발생하여 노드 F가 노드 A, B, D와 모두 비교하고 일치하는 노드를 찾지 못한 것을 나타내고 있다.

$$11011011 \wedge 01000011 = 01000011 \qquad (4)$$

본 논문에서는 허위 통과의 발생률을 낮추

기 위해서 <그림 6>과 같이 기준 경로의 노드를 두 개의 집합으로 나누고, 두 노드 집합에서 bit-wise OR 연산을 통해 각각의 블록 시그니쳐를 구성한다. 이를 통해 블록 시그니쳐를 구성하는 비트 패턴에 1의 수가 줄어들기 때문에 허위 통과의 발생률을 줄일 수 있다.

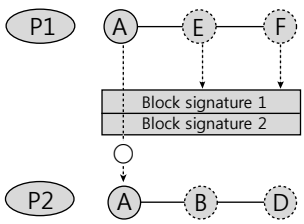


〈그림 6〉 이중 블록 시그니쳐

식 (5)~식 (6)과 같이 대표 경로의 노드 F를 기준 경로의 두 블록 시그니쳐와 bit-wise AND 연산을 수행한 결과, 대표 경로의 노드 F와 기준 경로의 두 블록 시그니쳐 모두 일치하지 않기 때문에 기준 경로에는 노드 F가 존재하지 않다고 판단할 수 있다. 만약 하나라도 같게 된다면 동일한 노드가 존재하는 것이다.

$$11011010 \wedge 01000011 \neq 01000011 \quad (5)$$

$$01001001 \wedge 01000011 \neq 01000011 \quad (6)$$



〈그림 7〉 이중 블록 시그니쳐 사용

<그림 7>의 이중 블록 시그니쳐를 사용했을 때 노드 F가 블록 시그니쳐 필터에 걸러져 P2의 모든 노드와 비교하지 않는 것을 보여준다.

4. 성능 평가

이 장에서는 본 논문에서 제안한 기법을 사용했을 경우와 기존 방법을 사용했을 경우의 성능을 비교 평가한다. 제안한 기법은 Java로 구현했으며 Microsoft Windows 7 운영체제가 설치된 2.50GHz CPU(Intel® Core™2 Quad)와 3GB RAM을 가진 PC 환경에서 실험을 수행하였다.

XML 문서는 작성자의 의도에 따라 다양한 구조와 용어가 사용되기 때문에 서로 다른 문서의 경로 간 노드를 비교하면 노드 간 서로 일치하지 않는 경우가 많이 발생된다. 이와 같은 XML 특성을 고려하여 본 논문의 제안 기법을 사용할 경우, 기존 기법을 사용했을 때보다 노드 비교 성능의 우수함을 증명하기 위해 다양한 구조와 용어가 포함된 30개의 XML 문서를 추출하였다. 이 문서들에 포함된 경로 수는 235개, 각 문서 당 평균 경로 수는 8개, 각 경로 당 평균 노드 수는 3개이다. 성능평가는 경로 간 노드 비교 수, 시그니쳐 기법으로 인해 발생하는 허위 통과의 발생 수, 경로 간 노드 비교 시간, 블록 시그니쳐와 경로 시그니쳐의 비교 수를 측정하여 제안 기법과 일반적인 방법을 비교 분석하였다.

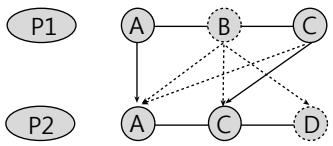
앞으로 나오는 성능 평가 그래프에 표시된 제안 기법 1, 제안 기법 2, 일반 기법의 의미

는 다음과 같다. 제안 기법 1은 본 논문에서 제안한 경로 시그니쳐를 이용한 기법으로 하나의 블록 시그니쳐를 이용한 것이고, 제안 기법 2는 이중 블록 시그니쳐를 이용한 기법이다. 그리고 일반 기법은 경로 시그니쳐를 이용하지 않고 일치하는 노드를 찾을 때까지 비교한 것이다. 경로 사이의 유사도는 Yoon et al.[12]에서 제안한 의미적 유사도를 고려한 측정 방법을 이용하였고, 노드 비교의 수행 시간은 경로 추출 및 데이터베이스에 저장하는 시간을 제외한 노드 비교를 수행하는 시간만을 고려하였다. 단, 제안 기법은 경로 시그니쳐와 블록 시그니쳐를 구성하는 시간을 포함하였다. 실험 방식은 30개의 문서를 차례로 저장하여 클러스터링을 수행할 때, 각 문서 별 경로와 클러스터 대표경로 간 노드 비교 횟수, 노드 비교 시간과 누적 비교 횟수, 누적 비교 시간을 측정하였다.

4.1 경로 간 노드 비교 수 평가

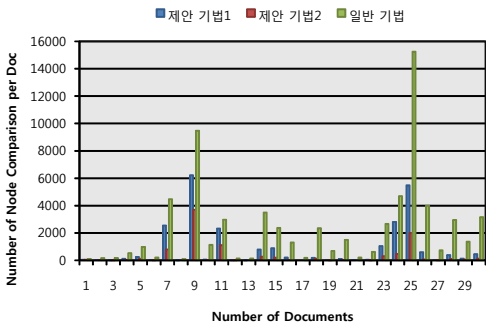
이 절에서는 본 논문에서 지정한 30개의 문서를 순차적으로 저장했을 때, 각 문서에서 추출된 경로의 노드와 대표 경로의 노드의 비교 횟수, 비교에서 실제로 매치된 노드의 비교 수와 매치되지 않은 노드의 비교 수를 측정한 것을 논의한다.

먼저 노드의 비교 수란 다음과 같다. <그림 8>에서 실선으로 된 화살표는 노드 간 매치가 된 관계를 나타내며 점선으로 된 화살표는 노드 간 비교 시 매치되지 않은 관계를 나타낸다. 즉, 실선으로 된 화살표의 수가 매치된 노드의 비교 수가되고 점선으로 된 화살표의 수가 매치되지 않은 노드의 비교수가 된다. 그리고 두 수를 합한 것이 경로 간 전체 노드의 비교수가 된다.

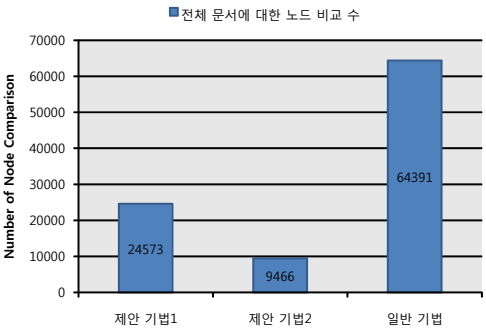


〈그림 8〉 경로 간 노드의 비교

<그림 9>는 문서 별 노드의 비교 수를 나타내고, <그림 10>은 전체 문서에 대한 노드의 비교 수를 나타낸다. 경로 간 노드의 비교 수는 제안 기법 2, 제안 기법 1, 일반 기법 순으로 모든 문서에 대해 일반 기법에서 가장 많은 노드 비교가 발생한 것을 알 수 있다.



〈그림 9〉 문서 별 노드 비교 수

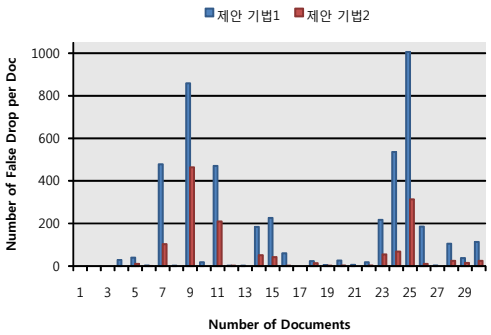


〈그림 10〉 전체 문서에 대한 노드 비교 수

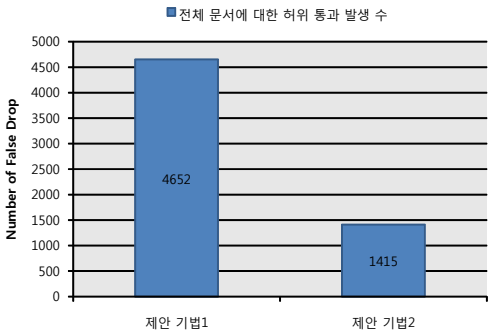
제안 기법 1과 제안 기법 2의 노드 비교 수는 일반 기법의 비교 수에 비해 각각 약 62%, 85%가 감소되었고, 허위통과 발생률을 줄이기 위해 제안된 제안 기법 2의 비교 수는 제안 기법 1의 비교 수에 비해 약 61%가 감소되었다.

4.2 허위 통과 발생 수 평가

이 절에서는 하나의 블록 시그니쳐와 이중 블록 시그니쳐를 이용했을 때 허위 통과의 발생 수를 비교 실험한 결과를 보인다. <그림 11>은 문서 별 허위 통과의 발생 수를 나타낸다. 이를 통해 모든 문서에 대해 제안 기법 2가 제안 기법 1에서 발생하는 허위 통과의 수를 줄인 것을 알 수 있다. 제안 기법 2는 블록 시그니쳐를 두 집합으로 분류함으로써 각 블록 시그니쳐를 구성하는 비트 패턴에 1의 개수를 줄였기 때문에 허위 통과의 발생 수가 줄어든 것이다. 그리고 <그림 12>는 전체 문서에 대한 허위통과 발생 수를 나타낸 것으로, 제안 기법 2에서 발생한 허위 통과 수는 제안 기법 1의 약 30%에 해당한다. 즉, 이 결과로 제 4.1절에서 확인한 것과 같이 노드 비교 수가 줄어든 것을 알 수 있다.



〈그림 11〉 문서 별 허위 통과 발생 수

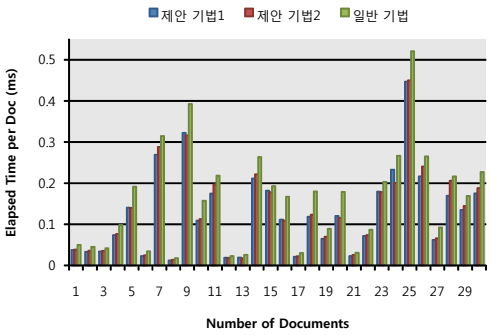


〈그림 12〉 전체 문서에 대한 허위 통과 발생 수

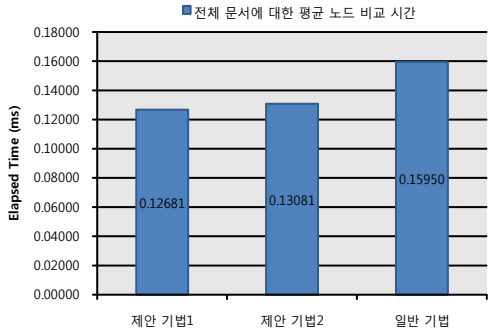
4.3 경로 간 노드 비교 시간 평가

제 4.1절에서 경로 간 노드 비교에 경로 시그니쳐를 적용했을 때 노드의 비교 수가 줄어든 것을 확인하였고, 제 4.2절에서는 제안 기법 2가 허위통과의 발생률을 줄인 것을 확인하였다. 이 절에서는 앞 절에서 확인한 결과로 실제 비교 수행 시간이 향상되었음을 증명한다.

<그림 13>은 문서 별 노드 비교 시간을 보여주고 있다. 모든 문서에 대해 제안 기법 1과 제안 기법 2 모두 일반 기법보다 빠른 수행 시간을 보이고 있다. 그리고 제 4.1절에서 노드의 비교 수 측정 결과와는 달리 21개의 문서에 대해 제안 기법 1이 제안 기법 2보다 빠른 수행시간을 보였다. <그림 14>는 전체 문서에 대한 평균 노드 비교 시간으로 제안 기법 1과 제안 기법 2의 수행 시간은 일반 기법의 수행 시간보다 각각 약 20%, 18%씩 단축되었고 제안 기법 1이 가장 좋은 성능을 보인다. 이는 일반 기법이 1분이 걸렸을 때,



〈그림 13〉 문서 별 노드 비교 시간

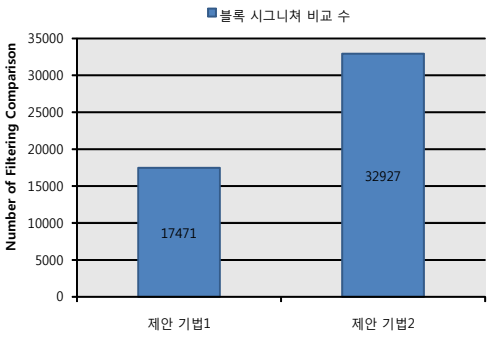


〈그림 14〉 전체 문서에 대한 평균 노드 비교 시간

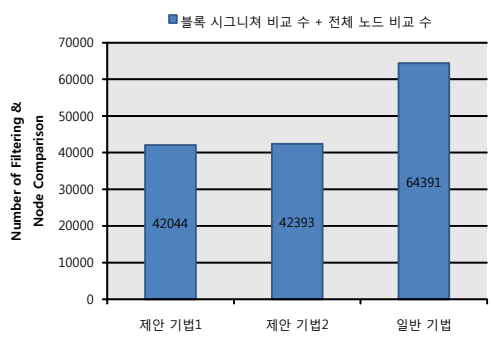
제안 기법 1과 제안 기법 2에서는 각각 48초, 50초가 걸리는 것으로 볼 수 있다.

4.4 블록 시그니쳐 비교 수 평가

제 4.1절과 제 4.2절에서 제안 기법 2가 제안 기법 1보다 비교 수와 허위 통과의 발생 수가 적은 것을 확인하였다. 그러나 제 4.3절에서 노드의 비교 시간은 제안 기법 1이 빠른 수행 시간을 보였다. 그 이유는 제안 기법 2가 허위 통과를 줄이기 위해 이중 블록 시그니쳐를 사용했기 때문인데, <그림 15>와 같이 블록 시그니쳐 비교는 제안 기법 1보다 제안 기법 2에서 약 2배 이상 많이 발생하였다.



〈그림 15〉 전체 문서에 대한 블록 시그니쳐 비교 수



〈그림 16〉 전체 문서의 저장에 대한 블록 시그니쳐 비교 수와 노드 비교 수

그래서 블록 시그니쳐 비교 수와 노드 비교 수를 합하면 전체 비교 수가 되고, 전체 비교 수는 <그림 16>과 같이 제안 기법 1이 약간 적은 비교 수를 나타낸다. 이를 통해 노드 비교 수를 줄인 것보다 허위통과를 줄이기 위한 오버헤드가 더 크다는 것을 알 수 있다.

5. 결 론

본 논문에서는 경로의 클러스터링을 수행하기 위해 데이터베이스에 저장할 XML 문서의 경로와 클러스터 대표 경로를 비교할

때 발생하는 불필요한 노드 비교를 최소화하는 기법을 제안하였다. 그래서 경로 클러스터링을 위해 두 경로 간 노드의 비교를 수행하기 전, 두 경로에 모두 존재하지 않는 노드인지를 판별하고 모두 존재하는 노드가 아니면 필터링하여 비교 횟수와 수행 시간을 줄였다. 또한, 시그니쳐 기법으로 인해 나타나는 허위 통과의 발생률을 줄이기 위해 이중 블록 시그니쳐를 제안하여 허위 통과의 발생률과 노드의 비교 횟수를 줄였지만, 매번 각 노드 시그니쳐가 두 개의 블록 시그니쳐와 비교하는 오버헤드로 인해 하나의 블록 시그니쳐를 이용할 때보다 약간 낮은 성능을 보였다. 향후 연구로는 이중 블록 시그니쳐를 이용했을 때 발생하는 오버헤드 문제를 해결하기 위한 연구를 지속할 것이다.

참 고 문 헌

[1] 김우생, “비트벡터에 기반한 XML 문서 군집화 기법”, 전자공학회논문지 C1, 제47권, 제5호, pp. 10-16, 2010.

[2] 김현주, 박소미, 박석, “확장된 질의 처리를 위한 경로간 의미적 유사도를 고려한 XML 문서 순위화 기법”, 정보과학회논문지 D, 제37권, 제2호, pp. 113-120, 2010.

[3] 이경하, 문봉기, 이규철, “관계형 XML 가지 패턴 질의를 위한 비트맵 인덱스와 질의 처리 기법”, 정보과학회논문지 D, 제37권, 제3호, pp. 146-164, 2010.

[4] 이범석, 황병연, “XML 문서의 유사 경로 검색을 위한 인덱싱 시스템”, 정보처리학회 논문지, 제15-D권, 제2호, pp. 171-178, 2008.

[5] Dalamagas, T., Cheng, T., Winkel, K. J., and Sellis, T., “A Methodology for Clustering XML Documents by Structure,” Information Systems, Vol. 31, No. 3, pp. 187-228, 2006.

[6] Faloutsos, C., “Signature Files : Design and Performance Comparison of Some Signature Extraction Methods,” ACM SIGMOD, pp. 63-82, 1985.

[7] http://www.w3.org/TR/REC-xml/.

[8] Hwang, J. H. and Ryu, K. H., “Clustering and Retrieval of XML Documents by Structure,” Lecture Notes in Computer Science, Vol. 3481, 2005.

[9] Lee, J. M. and Hwang, B. Y., “Path Bitmap Indexing for Retrieval of XML Documents,” Lecture Notes in Computer Science, Vol. 3885, pp. 329-339, 2006.

[10] Sacks-Davis, R., Kent, A., and Ramamohanarao, K., “Multikey Access Methods Based on Superimposed Coding Techniques,” *ACM Transactions on Database Systems*, Vol. 12, No. 4, pp. 655-696, 1984.

[11] XQEngine, http://www.fatdog.com.

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

저 자 소 개

장경훈 (E-mail : sosiziya@catholic.ac.kr)
2010년 가톨릭대학교 컴퓨터공학과 (학사)
2012년 가톨릭대학교 컴퓨터공학과 (석사)
2012년~현재 NHN I&S 사내정보시스템센터
관심분야 SNS, 데이터마이닝, XML 데이터베이스

황병연 (E-mail : byhwang@catholic.ac.kr)
1986년 서울대학교 컴퓨터공학과 (학사)
1989년 한국과학기술원 전산학과 (석사)
1994년 한국과학기술원 전산학과 (박사)
1994년~현재 가톨릭대학교 컴퓨터정보공학부 교수
1999년 University of Minnesota 방문교수
2007년 California State University 방문교수
관심분야 SNS, XML 데이터베이스, 데이터마이닝, 정보검색, 지리정보시스템