순차 패턴을 이용한 XML문서의 유사성 계산 방법 분석

이원철[O] 이상민
강원대학교
woncheol@mail.kanwon.ac.kr[O], smrhee@cc.kangwon.ac.kr

Korea Information Science Society

lee won cheol Han[0] lee sang min
Dept. of Computer science Kangwon National University

요 약

XML 문서의 요소는 의미적인 정보와 트리기반의 구조적인 정보를 포함하고 있기 때문에 요소의 구조적인 유사성이 곧 XML 문서의 유사성으로 연구되어 왔다. 그러나 구조적이고 순차적인 유사성만을 고려한 순차패턴 유사성 검색 방법은 의미적인(sementic) 유사성을 제대로 반영을 할 수가 없다. 이것은 정보 검색에 있어 재현율(recall)을 낮을 수밖에 없는 원인을 제공한다. 따라서 본 논문에서는 기존에 사용되었던 순차패턴을 기반으로 한 유사성의 계산 방법과 각각의 연구 방법이 의미적인 유사성에 대하여 한계가 있음을 찾아보았다.

1. 서 론

XML 문서는 웹상에서 데이터 표현과 교환 수단으로 급속히 증가하고 있다. 이러한 XML 문서는 반구조적인 문서로 자유로운 태그 사용으로 정보 검색 및 활용에 많은 이점을 제공하고 있다. XML문서의 자유로운 확장성으로 많은 유사 문서를 만들 수 있게 되었다. 그러나 정보의 검색 및 데이터 통합을 위하여 문서사이의 유사성에 대하여 검사할 수 있는 방법이 필요하게 되었다. XML문서의 유사성에 대한 연구는 요소의 구조적이고 의미적인 정보를 주로 이용하여 유사성을 계산하였다.
XML 문서의 요소는 의미적인 정보와 트리기반의 구조적인 정보를 포함하고 있기 때문에 요소의 구조적인 유사성이 곧 XML 문서의 유사성으로 연구되어 왔다.
본 논문에서는 기존의 유사성 검사기법을 한계점을 제시하고자 한다. 기존의 연구는 각 요소의 의미간의 유사성과 구조간의 유사성을 순차패턴을 바탕으로 검출하였다. XML문서는 비순차적인 구조와는 다르게 의미가 부여된 엘리먼트의 순차적이고, 계층적인 구조로 이루어져 있기 때문이다. 그러나 구조적이고 순차적인 유사성만을 고려한 순차패턴 유사성 검색 방법은 의미적인(*sementic*) 유사성을 제대로 반영을 할 수가 없다. 이것은 정보 검색에 있어 재현율(recall)을 낮을 수밖에 없는 원인을 제공한다.
따라서 본 논문에서는 기존에 사용되었던 순차패턴을 기반으로 한 유사성의 계산 방법과 각각의 연구 방법이 의미적인 유사성에 대하여 한계가 있음을 찾아보았다.

2. XML 문서에 대한 기본 개념과 예제문서

순차 패턴 XML 문서 트리는 다음과 같이 정의 될 수 있다.

- T는 XML문서 트리의 이름으로 상징적인 튜플을 의미한다.
- V는 노드의 제한된 셋(set)이다.
- root는 트리 문서의 첫 번째 노드이다.
- 순서화된 레벨 트리(Ordered Labeled Tree)이다. 루트 트리는 자식 노드를 가지고 있으며 각각의 자식노드는 순서가 있고 레벨이 있다.
- 노드 레벨(*Node Label*)은 루트노드부터 각각의 자신 노드까지 연결된 정점(edges)의 개수이다.
- 노드 가지(Node Branch)는 한 노드의 자식 노드의 개수 이다.

예제 문서
다음의 XML 문서는 출판사 영진에 있는 책에 대한 XML 문서를 가지고 임의의 3가지 트리 구조로 표현하였다. 의미적으로 같은 내용이지만 구조적으로 다르게 표현 될 수 있다.
Tree T_b은 isbn을 기준으로 트리를 만든 것으로 기본 트리 구조이다. 이것을 변형하여 Tree T_1과 Tree T_2를 만들어 비교하여 보았다.

```
<book>
<isbn>8973811800</isbn>
<title>동백꽃</title>
<publish>소담출판사</publish>
<author>김유정</author>
<category>한국소설</category>
<keyword>동백꽃</keyword>
</book>
```

그림 1 XML 문서의 예제

위의 예제 문서로 다음과 같은 트리 구조를 가진 XML 문서를 작성할 수 있다.

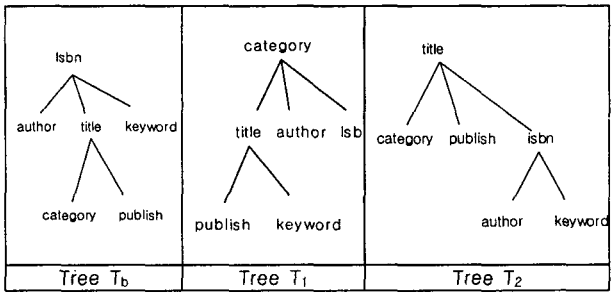

그림2 XML 문서의 트리

위의 예제문서로 기존의 순차 패턴 연구 방법을 분류하여 두문서의 유사성을 측정하였다.

3. 순차 패턴 유사성 계산

3.1. 가장자리 제한 방법(edge constraints)

유사성을 계산하기 위하여 선처리 과정으로 각 요소를 약자로 표현하였다.

isbn →i, title→t, publish→p, author→a, category→c, keyword→k로 간단하게 표현을 하였다. 다음과 같은 패스 셋을 구성할 수 있다.

e^{t_b} ={i→a, i→t, i→k, t→c, t→p}

e^{t_1} ={c→t, c→a, c→i, t→p, t→k}

e^{t_2} ={t→c, t→p, t→i, i→a, i→k}

두 문서 트리 사이의 같은 패스 셋을 이용하여 계산을 할 수 있다. 식은 다음과 같이 표현 할 수 있다.두 트리 문서의 전체적인 합집합에서 교집합의 비율을 계산하는 방법이다.

$$\text{sim}(T_b, T_1) = \left| \frac{e^{T_b} \cap e^{T_1}}{e^{T_b} \cup e^{T_1}} \right| = \frac{1}{9}$$

$$\text{sim}(T_b, T_2) = \left| \frac{e^{T_b} \cap e^{T_2}}{e^{T_b} \cup e^{T_2}} \right| = \frac{5}{6}$$

두 문서 트리는 T_b의 기본 문서의 트리에 대하여 같은 내용을 가지고 있지만 T_1, T_2의 트리 구조가 다르기 때문에 문서의 유사성을 계산하는 수식의 결과는 많은 차이가 있다.

3.2 패스 제한 방법(Path constraint)

패스 제한 방법은 주어진 순서 레벨을 처음부터 끝까지 연결하여 교집합을 계산하는 방법이다. 예를 들어 Tree T_b는 루트 노드에서부터 마지막 리프노드까지 연결하여 교집합을 계산한다.

Tree T_b,T_1,T_2 다음과 같은 패스 제한 셋을 가지게 된다.

p^{T_b}={i→a, i→t→c, i→t→p, i→k}

p^{T_1}={c→t→p, c→t→k, c→a, c→i}

p^{T_2}={t→c, t→p, t→i→a, t→i→k}

$$\text{sim}_p(T_b, T_1) = \frac{|p^{T_b} \cap p^{T_1}|}{|p^{T_b} \cup p^{T_1}|} = \frac{0}{8}$$

$$\text{sim}_p(T_b, T_2) = \frac{|p^{T_b} \cap p^{T_2}|}{|p^{T_b} \cup p^{T_2}|} = \frac{0}{8}$$

유사성의 계산 결과는 교집합이 전혀 존재하지 않기 때문에 두 문서 사이의 유사도는 계산을 할 수 없게 된다.

3.3 선형 측정 방법(Linear metric method)

전통적인 유사성 측정 방법은 문자열을 선형공간에 맵핑하는 방법으로 사전편찬상의 순서로 클러스터링 된 유사성을 측정하는 방법이다. 문자열을 사용하는 이유는 XML 문서는 문자열로 이루어졌기 때문이다. 문자열의 사이즈를 α라고 하면 선택된 정수 $\beta > 2\alpha$이다. 길이 n의 문자열을 $S_1 S_2 \ldots\ldots\ldots S_n$이라고 한다면 각각의 S_i는 1과 α사이의 t_i로 매핑이 될 수 있다. 따라서 전체 문자열은 $t_1/\beta + t_2/\beta^2 + \ldots\ldots\ldots + t_n/\beta^2$ 으로 표현할 수 있다.

일반적으로 $\beta = 2\alpha + 1$ 로 사용한다. α는 노드 레벨의 수이다.

XML 문서에서 패스 제한인 $p_i = v_1 \rightarrow v_2 \rightarrow \ldots\ldots\ldots \rightarrow v_n$ 는 다음과 같이 매핑이 될 수 있다.

$$Q_X(P_i) = t_X(v_1)/\beta + t_X(v_2)/\beta^2 + \ldots\ldots + t_X(v_n)/\beta^n$$

각 노드 레벨을 사전편찬상의 순서로 매핑하면 다음과 같다.

t(a)=1, t(c)=2, t(i)=3, t(k)=4, t(p)=5, t(t)=6

따라서 α는 6이 된다. $\beta = 2\alpha + 1$ =13이 된다.

p^{T_b}={i→a, i→t→c, i→t→p, i→k}

p^{T_1}={c→t→p, c→t→k, c→a, c→i}

p^{T_2}={t→c, t→p, t→i→a, t→i→k}

계산은 다음과 같이 할 수 있다.

Tree T_b

$$Q_{T_b}(i \rightarrow a) = t(i)/\beta + t(a)/\beta^2 = 3/13 + 1/13^2$$

$$Q_{T_b}(i \rightarrow t \rightarrow c) = t(i)/\beta + t(t)/\beta^2 + t(c)/\beta^3 = 3/13 + 6/13^2 + 2/13^3$$

$$Q_{T_b}(i \rightarrow t \rightarrow p) = t(i)/\beta + t(t)/\beta^2 + t(p)/\beta^3 = 3/13 + 6/13^2 + 5/13^3$$

$$Q_{T_b}(i \rightarrow k) = t(i)/\beta + t(k)/\beta^2 = 3/13 + 4/13^2$$

Tree T_1

$$Q_{T_1}(c \rightarrow t \rightarrow p) = t(c)/\beta + t(t)/\beta^2 + t(p)/\beta^3 = 2/13 + 6/13^2 + 5/13^3$$

$$Q_{T_1}(c \rightarrow t \rightarrow k) = t(c)/\beta + t(t)/\beta^2 + t(k)/\beta^3 = 2/13 + 6/13^2 + 4/13^3$$

$$Q_{T_1}(c \rightarrow a) = t(c)/\beta + t(a)/\beta^2 = 2/13 + 1/13^2$$

$$Q_{T_1}(c \rightarrow i) = t(c)/\beta + t(i)/\beta^2 = 2/13 + 3/13^2$$

Tree T_2

$$Q_{T_2}(t \rightarrow c) = t(t)/\beta + t(c)/\beta^2 = 6/13 + 2/13^2$$

$$Q_{T_2}(t \rightarrow p) = t(t)/\beta + t(p)/\beta^2 = 6/13 + 5/13^2$$

$$Q_{T_2}(t \rightarrow i \rightarrow a) = t(t)/\beta + t(i)/\beta^2 \rightarrow t(a)/\beta^3 = 6/13 + 3/13^2 + 1/13^3$$

$$Q_{T_2}(t \rightarrow i \rightarrow k) = t(t)/\beta + t(i)/\beta^2 \rightarrow t(k)/\beta^3 = 6/13 + 3/13^2 + 4/13^3$$

트리의 각각의 패스를 숫자 공간으로 매핑을 한 후 두 트리X 와 트리Y의 유사성을 두 쌍의 유클리안 디스턴스(Euclidean distance)의 평균을 계산할 수 있다.

$$\text{sim}_{inc}(X, Y) = 1 - \sqrt{\sum_{i=1}^{m} |Q_X(p_i) - Q_Y(p_i)|^2}$$

$\text{sim}_{inc}(T_b, T_1)$=0.6873

$\text{sim}_{inc}(T_b, T_2)$=0.1015

유사성의 계산 결과는 의미적 많은 차이를 발생함으로 결과 적으로 서로 유사성이 작은 것으로 판명이 된다. 따라서 의미적 유사성은 반영이 되지 않는다.

3.4 비용측정 방법(costs metric method)

전통적인 비용 측정 방법은 두 문서간의 목적 트리를 생성하고 가장 작은 연산을 수행하여 목적 트리로 변형하는 것이다. 사용하는 연산은 cost(insert), cost(update), cost(delete)를 사용하였다. 또한 각각의 연산에 대하여 가중치(weight factor)를 주었다. 또한 최소비용으로 두 문서를 목적 트리로 만든 다음 그 비용의 총 합계를 구하는 방법이다. 가중치는 루트노드에서 가까울수록 비용이 많이 추가되는 것이 일반적인 방법이다.

식으로는 다음과 같이 표기할 수 있다.

$$\cos t(op_i(v_j)) = \frac{1}{\triangle}\left(\sum_{i=1}^{n} w_i o_i(op_i) + \sum_{i=1}^{n} w_j \theta_j(v_j)\right)$$

w_i, w_j 가중치

$\triangle$ 일반화 요소

$o_i(op_i)$ 특징이 반영된 연산자

$\theta_j(v_j)$ 특징이 반영된 노드값

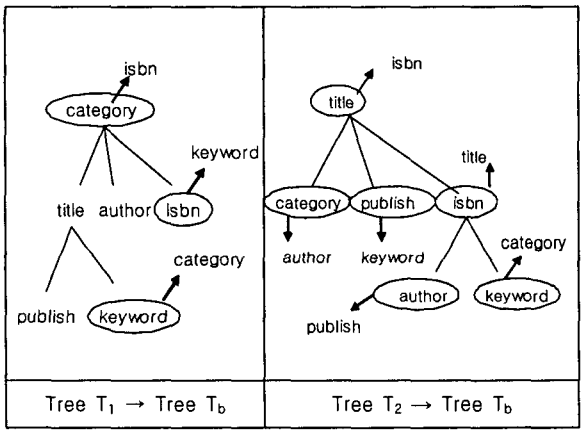

그림3 두 문서의 변형

Tree T_1을 Tree T_b로 변환하는 것과 Tree T_2를 Tree T_b로 변환하는 것에 대하여 계산하기위하여 다음과 같은 가중치 및 비용을 정의하였다.

o_1 = insert operation = 0.4

o_2 = delete operation = 0.4

o_3 = update operation = 0.6

θ_1 = node level = 0.5

θ_2 = node branch = 0.5

θ_3 = sementic interpretation = 0

$\triangle$ = 6

w_i, w_j = 가중치 = 1

목적 트리로 변경하기 위하여 두 개의 트리를 업데이트를 수행하였다. 의미적으로 같고 같은 요소의 개수를 가지고 있기 때문에 다른 연산 비용은 필요하지 않는다.

먼저 루트 노드에 있는 것을 업데이트를 한 후 리프노드로 내려가며 업데이트를 수행하였다. Tree T_1의 경우 3개를 업데이트 하지만 Tree T_2는 모든 노드를 업데이트를 해야 한다.

두 문서간의 거리를 계산하는 식은 다음과 같다

$$d(T_b, T_1) = \frac{1}{\triangle}\left(\sum_{i=1}^{n} w_i o_i(op_i) + \sum_{i=1}^{n} w_j \theta_j(v_j)\right)$$

$$d(T_b, T_1) = \frac{1}{6}(1\times0.6 + 1\times0.6 + 1\times0.6 + 1\times1, 1\times0.5 + 1\times0.5^2)$$

$$d(T_b, T_2) = \frac{1}{6}(1\times0.6 + 1\times0.6 + 1\times0.6 + 1\times0.6 + 1\times0.6 + 1\times0.6 + 1\times1 + 1\times0.5 + 1\times0.5 + 1\times0.5 + 1\times0.5^2 + 1\times0.5^2)$$

결과

$d(T_b, T_1)$=0.592

$d(T_b, T_2)$=0.961

계산 결과 두 트리는 상관이 적은 것으로 나타난다.

3.5 클러스터링에 의한 방법(clustering metric method)

클러스터링에 의한 방법은 XML 문서의 구조간의 유사성을 판별하기 위하여 순차 패턴을 이용한다. 먼저 XML 문서의 요소에서 순차 패턴을 이용하여 손실 없는 시퀀스 정보를 얻어낸다. 대표적인 시퀀스에 대한 발생 빈도뿐만 아니라 발생 순서를 찾아낸다. 이 정보를 이용하여 클러스터링을 한다.

클러스터링에 대한 방법은 충분한 예제문서가 있어야 하므로 본 논문에서 계산은 제외하였다. 결과적으로 순차 패턴을 이용한 클러스터링 또한 의미적 정보를 반영하기가 어렵다는 사실을 유추해 볼 수 있다.

4. 결론

XML 문서의 태그의 자유로운 정의와 내포된 구조 정보는 정보 검색 및 문서의 관리 분야에 많은 이점을 제공을 하고 있다. 그러나 이러한 확장성은 문서의 통합 및 검색의 연구를 필요하게 되었다. XML 문서의 구조적이고, 순차적인 패턴을 이용한 XML 문서의 유사성은 의미적인 유사성을 충분히 반영하기가 어렵다.

본 논문에서는 기존의 순차패턴을 이용한 XML 문서의 유사성이 의미적인 유사성을 반영하지 못하는 것을 증명하였다.

앞으로 XML 문서의 의미적인 유사성을 반영하기 위한 방법을 수행하고자 한다.

※참고 문헌

1. zhongping zhang "similarity Metric for XML Documents" 2003
2. Andrew Nierman and H. V. Jagadish "Evaluating Structural Similarity in XML Documents" webdb, 2002.6
3. 이정원, 이기호, "유사성 기반 XML문서 분석 기법", 정보과학회 2002.6
4. Torsten Schelieder "Similarity Search in XML Data using Cost-Based Query Transformations" webdb, 2001
5. Paolo Ciaccia and Wilma Penzo " Adding Flexibility to structure Similarity Query on XML on XML Data" FQAS, 2002

6. 황정희, 류근호, "순차패턴에 기반한 XML 문서 클러스터링" 정보처리학회논문지 D 2002,12
7. 조정길, 구연설 "스키마가 없는 XML 문서에서의 재사용 가능한 XML schema 추출기법" 정보처리학회논문지D 2003,8