Journal of The Korea Society of Computer and Information
Vol. 22 No. 4, pp. 59-65, April 2017

www.ksci.re.kr
https://doi.org/10.9708/jksci.2017.22.04.059

Extraction of similar XML data based on XML structure and processing unit

Jong-Hyun Park *

Abstract

XML has established itself as the format for data exchange on the internet and the volume of its instance is large scale. Therefore, to extract similar information from XML instance is one of research topics but is insufficient.

In this paper, we extract similar information from various kind of XML instances according to the same goal. Also we use only the structure information of XML instance for information extraction because some of XML instance is described without its schema.

In order to efficiently extract similar information, we propose a minimum unit of processing and two approaches for finding the unit. The one is a structure-based method which uses only the structure information of XML instance and another is a measure-based method which finds a unit by numerical formula. Our two approaches can be applied to any application that needs the extraction of similar information based on XML data. Also the approach can be used for HTML instance.

▸ Keyword : XML mining, Similar XML data, Similar web data, Similar information retrieval

I. Introduction

인터넷의 급속한 성장과 보급으로 웹 상에 존재하는 정보의 양은 엄청나며, 그 종류 또한 매우 다양하다. 그러나 웹을 통해 접근 가능한 방대한 정보량에 비해 실제 사용자 개개인이 필요로 하는 정보는 극히 일부분인 경우가 대부분이다. 그러므로 현재 이를 정확하고 빠르게 얻기 위한 연구는 매우 활발히 진행되고 있으며, 유사 정보추출은 이러한 연구들 중 대표적인 하나이다[1]. 현재 인터넷 상의 많은 정보들은 HTML로 기술되어 있으며 HTML 문서로부터 유사정보를 추출하기 위한 연구는 매우 다양하게 진행되고 있다. 이와 함께 인터넷 상의 정보 교환을 위한 새로운 표준인 XML 데이터 역시 그 양이 급격히 증가하고 있으며, 이를 기반으로 유사 정보를 추출하기 위한 요구 역시 자연스럽게 증가하였다. 물론 HTML 문서로부터 유사 정보를 추출하기 위한 여러 방법들이 존재하므로 이들 가운데 일부를 유사 XML 데이터를 검색하기 위하여 앞선 방법들을 수용하거나 참고하여 사용할 수 있다. 그러나 HTML문서의 경우 구조정보는 주로 스타일 정보를 표현하고 있는 반면, XML문서의 경우 구조정보는 내용정보의 의미를 표현한다. 그러므로 현재 제안된 HTML문서의 유사정보 추출에 관한 연구들 대부분은 HTML문서의 구조정보를 유사정보 추출을 위해 사용한다고 보기는 힘들다. 그러나 XML 문서의 경우 구조정보를 유사정보 검색을 위해서 사용한다면 보다 효율적이고 정확한 검색이 가능하다.

XML은 현재 다양한 분야에서 사용 중에 있으며 또한 사용하고자 시도 중에 있다. 동일한 목적을 갖는 응용들 간에는 상호 교환을 위한 XML 문서의 구조를 미리 정의하고 정보를 공유하고자 한다. 그러나 동일한 목적을 갖는 모든 응용들이 동일 구조를 따르고 있지 못하는 것이 현실이며, XML문서의 사용이 일반화 되면서 때로는 문서의 구조를 미리 정의하지 않고 사용하는(DTD나 Schema 없는) 문서들도 존재한다. 이러한 현실은 유사 정보 추출의 측면에서 구조정보의 참조 없이 문서들 사이

• First Author: Jong-Hyun Park, Corresponding Author: Jong-Hyun Park
*Jong-Hyun Park (jonghyunpark@cnu.ac.kr), Dept. of Computer Engineering&Science, Chungnam National University
• Received: 2017. 03. 14, Revised: 2017. 03. 28, Accepted: 2017. 04. 14.
• This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government [NRF-2014R1A1A2057221].

의 구조정보를 판단하고 이를 기반으로 내용정보를 추출할 필요성이 대두 되었다.

본 논문에서는 동일한 목적을 갖지만 서로 다른 구조로 표현되는 XML 문서들로부터 문서의 구조정보 만을 이용하여 유사정보를 추출하고자 한다. 이를 위하여 유사정보검색을 위한 최소 처리단위를 제안한다. 처리단위의 결정은 XML 문서로부터 유사 정보 검색 시, XML 문서 내부에서 의미적으로 구분되는 문서의 일부분을 결정하는 것이다. 다시 말하면, 하나의 XML 문서가 의미적으로 여러 부분으로 구분될 수 있다면, 구분된 문서의 일부분이 처리단위가 된다. 우리의 처리단위는 HTML문서의 Table또는 List와 같은 의미 있는 내용을 포함하는 문서의 일부분이다. 처리단위를 결정하기 위해서 본 논문은 두 가지 방법을 제안한다. 한 가지는 XML문서의 구조적인 정보만을 이용하여 처리단위를 결정하는 구조기반 접근 방법이고 다른 한 가지 방법은 수식을 이용하여 처리단위를 결정하는 수치기반 접근방법이다. 우리는 두 가지 방법을 구현하여 결과가 동일한지 여부를 확인하고 어떤 방법이 어떤 특성을 갖는지를 정의한다.

본 논문의 목적은 다음과 같다.

- XML 데이터로부터 유사정보를 자동으로 추출하기 위한 알고리즘의 제안
- 유사정보 추출을 위한 처리단위를 정의하고 이를 결정하기 위한 두 가지 방법을 제안
- 처리단위 결정을 위한 두 가지 방법을 구현하고 성능을 평가

논문에서 제안하는 유사정보 추출 방법은 정보의 유사성을 판단하기 위해 구조적인 정보만을 사용함으로 구조정보가 없는 형태의 문서에도 적용이 가능하다. 그러므로 우리의 방법은 XML 문서뿐 만 아니라 약간의 변형만으로 HTML문서를 비롯해 다양한 반 구조적 문서를 위해서 적용이 가능하다.

본 논문의 나머지 구성은 다음과 같다. 2장에서는 유사정보 추출을 위해 이미 연구된 관련 연구들을 기술하고 있으며, 3장에서는 논 논문에서 제안하고 있는 유사 XML 데이터를 추출하기 위한 기본 아이디어와 유사 XML 데이터 추출을 위한 처리단위와 이를 결정하기 위한 두 가지 방법에 대해서 자세히 기술하고 있다. 4장에서는 처리단위를 결정하기 위한 두 가지 방법의 평가를 기술하고 있으며, 마지막으로 5장에서 본 논문의 결론과 향후 연구방향에 대해서 기술한다.

II. Related works

XML 문서는 HTML과 유사한 반 구조적 문서이다. 그러므로 HTML 문서로부터 유사한 정보를 얻기 위한 연구는 XML 문서의 유사 정보 추출과 직결된다. HTML 문서로부터 유사정보를 추출하기 위한 주 연구 흐름은 HTML 문서로부터 임의의 반복되는 패턴(이러한 pattern들은 마치 데이터베이스의 table과 유사하게 간주된다)을 찾기 위한 연구들이 있다[1, 2, 3, 4, 5]. 다시 말하면, 그들은 HTML 문서 안에 반복되는 패턴은 의미적으로 연관된 정보들을 기술한다고 가정하고, 이러한 패턴으로 기술된 정보들로부터 유사정보를 추출한다. 우리는 이러한 연구들로부터 반복되는 패턴의 아이디어를 XML 문서의 유사정보 검색을 위하여 적용한다. XML과 HTML의 가장 다른 점은 XML 문서는 일반적으로 문서의 구조에 의미적인 내용을 포함한다는 것이다. 이러한 정보는 HTML 문서로부터 유사정보를 추출하기 위한 방법보다 정확한 방법을 정의하기 위해서 사용될 수 있다. 그러므로 우리는 XML의 구조에 포함되어있는 정보를 이용하여 XML 문서로부터 유사정보를 추출하기 위한 방법을 제안한다.

XML 문서로부터 유사정보를 추출하기 위한 연구는 그리 많지 않으며 이와 유사한 접근 방법으로 XML 문서의 구조적인 유사성을 찾고자하는 다수의 연구들이 존재한다[6, 7, 8, 9]. 그러나 이러한 접근 방법을 유사 정보 검색을 위해서 그대로 사용하기는 힘들다. 다시 말하면, 두 XML 문서의 구조가 유사하다할 지라도 두 XML 문서가 기술하는 내용은 다를 수 있다. 예를 들어 '디지털 카메라'의 정보를 기술하기 위한 XML 문서와 '모니터'의 정보를 기술하기 위한 XML 문서의 구조는 유사할 수 있지만 그 내용은 다르다. 또한 XML 문서들 사이의 구조적 유사성을 찾기 위하여 DTD와 Schema 같은 정보를 이용한 접근 방법도 존재한다[10, 11]. 그러나 모든 XML 문서가 구조정보를 포함하고 있지는 않으므로 본 논문에서는 순수한 XML 문서로부터 유사한 정보를 얻기 위한 방법을 제안한다. 최근 온톨로지와 같은 지식정보 표현을 위한 표준들이 제안되면서 시멘틱 웹 환경에서 온톨로지를 이용하여 정보들 사이의 유사도를 추정하거나 추론을 통해 유사 정보를 추출하기 위한 연구들 또한 존재한다[12, 13]는 . 그러나 이러한 방법 역시 유사정보 추출을 위해 추가적인 정보를 활용하므로 본 논문의 목적과 일치하지는 않는다.

III. Extraction of similar XML data

유사 XML 데이터 검색은 사용자가 입력한 키워드를 이용하여 다양하게 존재하는 XML 문서로부터 해당 키워드와 유사한 정보를 추출하는 것을 그 목표로 한다. XML문서는 동일한 응용에서 사용할 경우, 대부분 동일한 구조를 따른다. 다시 말하면, 단일 도메인에 포함된 응용들은 상호운용을 위해서 동일한 XML 문서의 구조를 정의하여 사용한다. 그러나 동일한 목적을 띄지만 서로 다른 도메인에 포함된 응용들 간에는 서로 다른 문서의 구조를 정의하고 사용한다. 그러므로 사용자는 유사정보를 찾기 위해서는 각각의 응용에 존재하는 XML 문서의 구조정보를 참조하여야 한다. 설령 각각의 문서의 구조정보를 참조

한다 하더라도 각각의 구조정보에 대한 의미론적인 정보도 함께 이해를 하여야 유사정보를 찾을 수 있다. 또한 어떤 응용들에서는 데이터 타입 선언 없이 혹은 응용 내부에서만 사용하는 데이터 타입 선언을 따르는 XML 문서를 사용하기도 한다. 그러므로 본 논문에서는 XML 문서의 DTD혹은 Schema정보 없이 XML 문서(instance)로부터 유사정보를 추출하기 위한 방법을 제안한다.

1. Basic Idea

일반적으로 XML의 태그(tag) 이름과 구조는 포함하고 있는 자손 정보들의 의미를 포함하고 있다. 즉, 동일한 구조를 갖는 XML 노드는 유사한 의미를 갖는다고 가정할 수 있다. 그러므로 사용자가 입력한 키워드를 값으로 갖는 노드와 동일한 구조 정보를 갖는 노드는 유사한 정보일 확률이 높다. 유사정보를 찾기 위한 우리의 방법은 크게 두 가지 단계로 수행된다. 첫 번째 단계는 문서에서 키워드를 검색하여 키워드를 포함하고 있는 텍스트 노드를 검색하고 해당 노드의 구조정보를 추출한다. 두 번째 단계는 얻어진 구조정보를 이용하여 동일 구조를 갖는 노드를 찾아내고 그 값을 얻는다. 이렇게 얻어진 값은 첫 번째 단계의 입력인 키워드와 유사한 정보가 된다. 위의 두 단계는 새로운 유사 정보를 찾을 수 없거나 사용자가 원할 때 까지 반복한다.

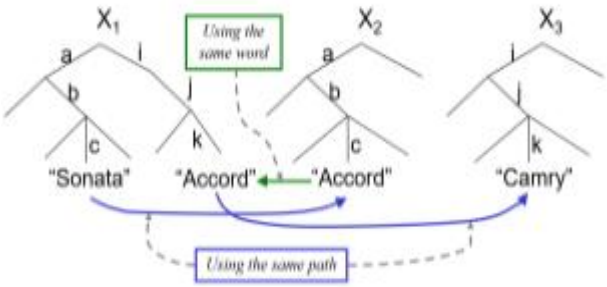

Fig. 1. The basic idea for extracting similar XML data

[그림 1]의 예제에서 사용자의 초기 입력 키워드는 'Sonata'이다. 이 키워드를 이용하여 X1문서로부터 'Sonata'를 값으로 갖는 노드를 찾을 수 있다. 찾아진 노드의 구조 정보는 'a/b/c'이다. 그러므로 이 구조정보를 이용하여 X2로부터 유사 워드인 'Accord'를 찾을 수 있다. 'Accord'는 새로운 키워드가 되고 X1에서 새로운 경로인 'i/j/k'를 찾을 수 있다. 새로운 구조정보 'i/j/k'는 다시 X3에 적용되어 'Camry'를 유사 정보로 검색한다. 유사 정보 검색을 위하여 필요한 또 하나의 입력은 동일한 구조 정보를 갖는 최소 키워드의 수(f) 이다. 위 [그림 1]의 경우 사용자의 입력 키워드는 하나이지만 사용자는 하나이상의 키워드를 입력으로 제공할 수 있다. 예를 들어 사용자가 3개의 키워드를 입력했을 경우, 2개의 워드에 해당하는 구조정보는 동일하지만 나머지 하나의 워드에 해당하는 구조정보가 다를 경우 검색된 구조정보를 선택할 것인지를 결정해야한다. 이때 f가 2라면 검색된 구조정보는 선택 되지만 3이라면 선택되지 않는다.

2. Processing Unit

많은 경우 XML 문서는 동일한 구조를 반복하여 사용한다. 다시 말하면, 한 문서에 동일한 구조의 하위 트리 구조를 여러 개 포함하고 있다. 이러한 경우, 각 하위 트리들은 구조와 의미적으로는 동일하지만 서로 다른 하위 트리에 존재하므로 유사한 정보라고 볼 수 없다. 그러므로 본 논문은 유사정보 검색을 위해 동일한 의미의 정보를 포함하고 있는 최소의 하위 트리를 결정해야 한다. 본 논문에서는 이를 처리단위(Processing unit)라고 부르며, [그림 2]는 처리단위의 필요성을 보인다. [그림 2]의 경우, 예제문서는 논문집이고 사용자는 공동저자들을 검색하기 위하여 키워드로 'A'와 'B'를 그리고 앞서 언급한 동일 구조를 갖는 키워드의 최소 수(f)로 2를 입력한다. 논문의 기본 아이디어로부터 우리는 'A'와 'B'의 공동 구조 정보인 'Section List/Section/Articles/Article/Author'를 얻을 수 있다. 그러므로 유사 정보를 검색하기 위하여 구조정보를 사용하여 다시 'C', 'E', 'F', 'G'를 유사 정보로 얻는다. 그러나 'E', 'F', 'G'는 사용자가 원하는 유사정보가 아니다. 즉, 사용자는 'A'와 'B'의 공동저자인 'C'만을 검색하기를 원할 것이다. 그러므로 우리는 위 [그림 2]의 점선 부분과 같은 유사정보 검색 처리를 위한 최소의 처리단위를 결정하고 사용해야만 한다.

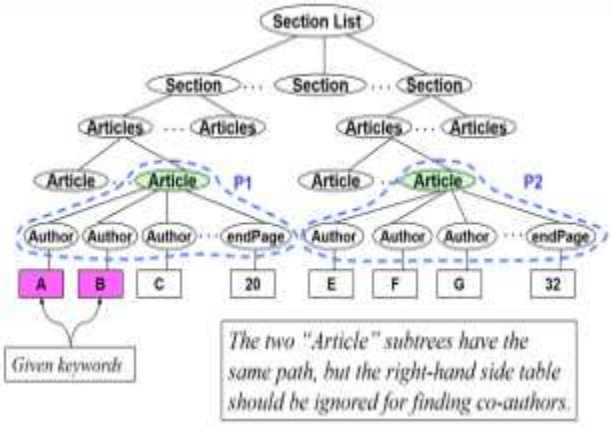

Fig. 2. The requirement of processing unit

사용자의 질의어는 단순히 하나의 키워드로 구성되는 것이 아니고 필요한 경우 의미적으로 연관된 여러 개의 키워드 집합이 하나의 질의어가 될 수 있다. 예를 들면 'Hyundai'와 'Sonata'라는 두 개의 키워드 집합이 하나의 질의어가 될 수 있다. 이 경우 'Sonata'는 자동차 브랜드로서의 의미를 갖는다고 예측할 수 있다. 본 논문에서는 하나의 키워드만을 질의어로 사용하는 경우를 단순 질의 라고 표현하고 둘 이상의 키워드로 구성된 질의어를 복합 질의라고 정의한다. 복합 질의의 경우 역시 처리단위의 개념은 단순 질의와 동일하다. 다만 복합 질의의 경우 고려해야할 점은 단순히 질의어에 포함된 키워드들뿐만 아니라 키워드들 사이의 연결 구조 역시 질의어에 포함된다는 것이다. 다시 말하면 XML 문서를 하나의 트리 구조로 볼 때, XML 문서를 구성하는 많은 서브 트리들 중 사용자가 질의한 키워드들을 모두 포함한 최소의 서브트리가 바로 복합질의 검

색단위이다.

[그림 3]은 자동차회사와 자동차회사별 자동차명을 보여주는 XML 문서의 한 예이다. 이 문서에서 점선으로 표시된 부분에서 보는 바와 같이 'Hyundai, Sonata'는 서로 다른 위치정보를 가지면서 서로 연관된 정보이다. 이때 점선 안 에 포함된 서브트리가 바로 복합 검색을 위한 질의어의 단위이다. 결국 복합 질의의 경우 입력 질의어의 단위는 서브트리이고 결과 역시 이와 동일한 구조 정보를 갖는 서브 트리들이다.

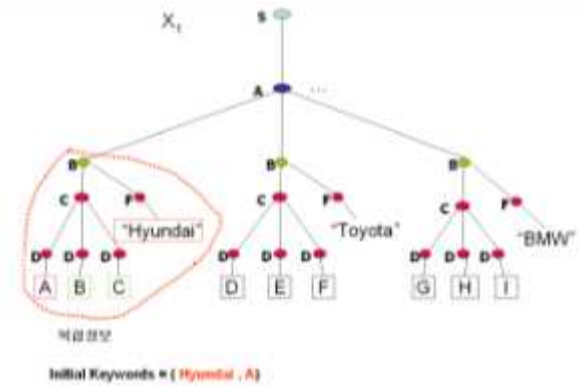

Fig. 3. Mixed search

본 논문에서의 처리단위 개념은 HTML의 TABLE또는 LIST와 유사한 개념으로 가장 유사한 정보가 포함된 최소의 단위이다. 또한 유사정보 검색을 위한 입력 키워드는 동일한 처리단위에 속해야만 동일한 구조 정보를 갖는다. 처리단위를 결정하기 위하여 본 논문에서는 두 가지 방법을 제안하고 있다. 첫 번째 방법은 XML 문서의 구조정보를 이용하여 처리단위를 결정하는 구조기반 접근 방법이고, 두 번째 방법은 계산을 통해 처리단위를 결정하는 수치기반 접근 방법이다.

2.1 Structure-based approach

구조기반 접근 방법은 처리단위를 결정하기 위하여 XML문서의 구조 정보만을 이용하는 방법으로 [그림 4]와 같이 정의한다.

- Let T is a tree, n is a node in T, and PU is Processing Unit.
- L(n) denotes the set of leaves of the subtree determined by n.
- Let W is a set of input words, f is the minimum number of keywords which have to include in PU.

n is the root node of PU in T with respect to W and f iff (1) $\#\{l \in L(n) \mid value(l) \in W\} >= f$
(2) $l_1, l_2 \in L(n)$, $value(l_1)$, $value(l_2) \in W$
-> $path(l_1)=path(l_2)$
(3) n is the deepest node that satisfies (1) and (2)
(4) if n = null then n is root node

Fig. 4. The definition of structure-based approach

처리단위는 XML 문서 T의 서브 트리 PU 이다. 그러므로 처리단위를 찾는 것은 서브트리의 최상위 노드 n을 찾으면 자연스럽게 문제가 해결된다. 처리단위를 찾기 위한 첫 번째 단계(1)는 사용자가 입력한 하나의 키워드 w를 XML 문서의 말단 노드로부터 찾는 것이다. 처리단위의 최상위 노드의 후보 노드는 키워드를 포함한 텍스트 노드의 조상 노드들이 된다. 처리단위를 찾기 위한 다음 과정(2)은 검색된 텍스트 노드로부터 최상위 노드까지 부모 노드를 찾아가면서 자손 노드들 가운데 사용자가 입력한 또 다른 키워드를 값으로 갖는 텍스트 노드가 존재하는지 여부를 찾는다. 만약 찾아 지면 해당 노드가 앞서 검색한 첫 번째 키워드가 검색된 노드의 경로와 동일한지 여부를 확인한다. (3) 만약 이러한 두 조건을 만족한다면 해당 텍스트 노드들의 조상 노드들 가운데 가장 경로가 짧은 부모 노드가 바로 처리단위의 최상위 노드가 된다. 물론 이 경우 처리단위에 포함된 검색된 키워드의 수는 f보다 크거나 같아야한다. 구조기반 접근 방법의 경우 검색된 처리단위가 문서 전체인 경우가 존재한다. 예를 들어, [그림 2]에서 사용자가 입력한 초기 키워드가 'A'와 'E'인 경우 처리단위는 문서 전체이다. 이러한 경우 구조기반 접근 방법은 모든 'Author'들을 검색하여 반환할 것이다. 물론 처음부터 사용자가 원한 결과가 'A'와 'E'가 함께 저자로 작성한 학술정보의 또 다른 공동 저자들의 검색 결과를 원한 것이 아니고, 'A'와 'B'가 저자로 있는 학술분야의 모든 저자들의 검색을 원했을 수도 있다. 그러나 구조기반 접근 방법의 경우 공동저자를 검색하기는 어렵다. 이러한 단점을 보완하기 위하여 본 논문은 사용자가 처리단위의 깊이를 결정할 수 있는 방법인 수치기반 접근 방법을 제안한다.

2.2 Measure-based approach

수치기반 접근 방법은 처리단위의 결정을 위한 또 하나의 방법으로 처리단위를 결정하기 위하여 XML 문서에 존재하는 노드에 특정한 가중치를 부여하고 이를 기반으로 처리단위를 결정하는 방법이다. 이러한 방법은 일반적인 경우 구조기반 접근 방법과 유사한 처리단위의 결과를 얻지만 필요에 따라 수치를 조정하여 처리단위의 크기를 조절할 수 있으므로 앞선 구조기반 접근 방법에서 발생할 수 있는 전체 문서가 처리단위가 되는 단점을 보완할 수 있다. [그림 5]는 수치기반 접근 방법의 정의이다.

K(n)은 처리 단위에 포함된 키워드의 총 개수이다. 결국 유효한 K(n)은 사용자가 입력한 f보다는 크거나 같아야 한다. f보다 큰 값을 갖는 K(n)을 찾으면 이때의 모든 n은 후보 처리단위의 최상위 노드가 된다. 그러면 이 후보 n들 중 어떤 n을 처리단위의 최상위노드로 결정해야할까 하는 문제가 남는다. D(n)은 처리단위의 최상위 노드에서부터 해당 키워드들까지의 깊이의 평균이다. U(n)은 서브트리에 포한된 키워드 수(K(n))를 D(n)으로 나눈 값이며, K(n)이 f보다 작은 경우 그 값은 0이다. 결국 U(n)은 최소 f개 이상의 키워드를 모두 포함하고 말단 노드에서부터 가장 가까운 노드이다. [그림 6]은 위 수식들을

간단히 설명하기 위한 그림이다. D(n)은 n을 최상위 노드로 하는 서브 트리에 포함된 키워드들의 깊이 평균을 의미 한다. U(n)은 서브트리에 포한된 키워드 수(K(n))를 D(n)으로 나눈 값으로 이때의 K(n)은 f보다 크거나 같아야 한다. 그리고 U(n)의 값 가운데 가장 큰 값을 갖는 U(n)의 n이 후보 처리단위의 최상위 노드가 된다. 만약 여러 개의 후보 처리단위들이 존재한다면 D(n)의 값이 가장 작은 것을 선택한다.

- w : keyword.
- W : keyword set.
- n : candidate Node for Processing Unit.
- N : Candidate Node Set for Processing Unit.
- f : The minimum number of Keywords, We can obtain f by user.
- $K(n) = \#\{w \in W \mid w \in \text{subtree}(n)\}$, The number of keywords contained in the n.
- $D(n) = \text{average}\{\text{depth}(w,n) \mid w \in W, w \in \text{subtree}(n)\}$, Depth From n to w.
- $U(n) = \begin{cases} K(n)/D(n) & \text{if } K(n) >= f \\ 0 & \text{otherwise} \end{cases}$
- A node n is a candidate processing unit, if $U(n) >= U(n_i)$, $\forall n_i \in N$.
- n is Processing Unit, If n is candidate processing unit and $D(n)=<D(n_i)$ for any candidate processing unit.

Fig. 5. The definition of measure-based approach

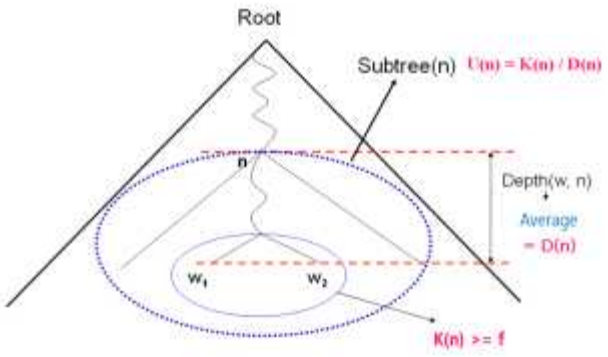

Fig. 6. The processing nuit by measure-based approach

[그림7]은 유사 정보 검색을 위하여 앞에서 정의한 수치기반 접근 방법을 적용하여 처리단위를 결정하는 예이다. 사용자가 입력한 초기 키워드 집합이 (A, B, C)이고, 선택요소, 즉 처리단위에 포함 되어야 할 최소 키워드의 수 f는 2로 지정 되었을 경우, 수치기반 접근 방법을 적용하는 첫 단계는 주어진 각 키워드에 이르는 경로를 찾는 것이다. 예제에서는 'A', 'B', 'C'의 값을 갖는 노드에 이르는(점선으로 표시된) 총 3개의 경로를 찾을 수 있다. 처리단위를 찾기 위한 다음 단계는 찾은 경로를 따라가면서 경로에 위치한 노드 'n' 에 대한 K(n)값과 D(n)값을 구하는 것이다. [그림7]의 경우, 'A'와 'B'키워드 정보를 포함하고 있는 'Article' 노드는 포함하고있는 키워드의 수인 2를 K(n)의 값으로 결정한다. 그리고 D(n)의 값으로 'Article' 노드로부터 각 키워드 노드에 이르는 깊이(depth)의 평균인 2를 결정한다. 결국 'Article' 노드의 U(n)은 1이다. 'C' 키워드의 경우, 해당 값을 포함하고있는 'Article'노드는 입력 정보로 정의한 선택요소의 값 f가 2이므로 K(n)의 값은 1이다. 하지만 1은 선택요소 조건에 만족하지 못하므로 U(n)값은 0이 된다.

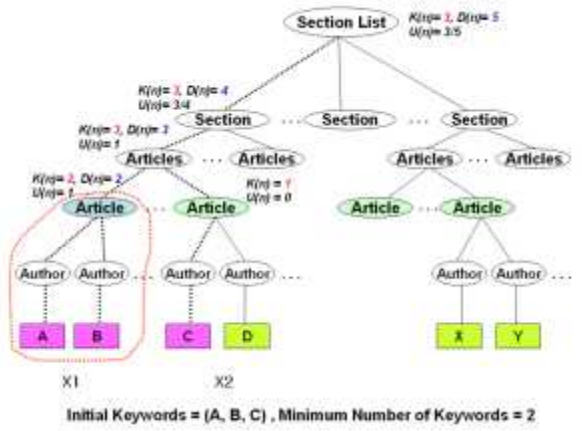

Fig. 7. The processing nuit by measure-based approach

위 예제에서 수치기반 접근 방법에 의해서 얻어진 처리단위는 첫 번째 'Article' 노드이고 이것은 구조기반 접근 방법으로 얻을 수 있는 처리단위와 동일하다.

- Input : T : tree
  W: a set of keywords
  f : the minimum number of keywords which have to include in Processing Unit
- Output : V : a set of words

```
Begin
  Find PU in T with W & f
    Find the path p of all leaves from the root of PU
    V={value(l) | path(the root of PU, l)=p}
  Return V
End
```

Fig. 8. The algorithm for finding similar XML data

유사정보를 찾기 위한 다음 단계는 찾아진 처리단위를 이용하여 유사 XML 데이터를 추출하는 것이다. [그림 8]은 유사 데이터를 찾기 위한 알고리즘으로 해당 알고리즘은 처리단위를 찾는 방법과는 독립적이다. 먼저 키워드 집합 W와 사용자가 찾고자하는 최소 키워드의 수 f를 기반으로 XML 문서 T에서 [그림 5]의 K(n)을 찾는다. K(n)이 찾아지면 D(n)과 U(n)을 찾아서 가장 큰 값을 갖는 U(n)의 n을 처리단위의 최상위 노드로 선택한다. 그 후 처리단위 내의 키워드와 동일한 경로의 값들은 새로운 키워드가 되고 이러한 과정은 더 이상 새로운 V를 찾을 수 없을 때까지 반복한다. 해당 알고리즘의 기본 시간 복잡도 O(n)는 'W * XML 문서의 총 말단 노드 수' 이지만 반복에 의해서 새로운 키워드가 얼마나 발견될지에 의존적이다.

IV. Evaluation

첫 번째 평가는 실제 의미 있는 유사 정보들이 추출되는가 하는 것이다. [표 1]은 논문집 XML 문서를 대상으로 'Ji-Hoon Kang'과 'Man-Ho Lee'를 질의어로 해당 저자들이 포함된 유사 데이터를 검색한 결과를 보인다. 대상이된 XML 문서는 DBLP의 많은 학술 자료들 가운데 1990년부터 2015년까지의 데이터베이스 분야의 논문집, 그리고 DBLP와는 XML 구조가 다른 sigmod 논문집을 대상으로 했으며 실험을 위한 f는 2이다.

Table 1. Similar XML data search by iteration

#Iteration	number of new XML data searched after iteration	The cumulative number of XML data
1	6	8
2	4	12
3	3	15
4	2	17
5	0	17

실험 결과 5회의 반복 검색이 진행되었으며 최종적으로 17개의 유사 데이터를 찾았다. 각 단계마다 검색된 데이터들은 초기 키워드들과 함께 새로운 질의어로 다음 단계의 입력이 되었으며 수작업으로 결과를 분석한 결과와 동일한 결과를 얻었다. 그러나 검색해야할 데이터의 집합이 크고 질의어의 중복이 있는 경우 원하지 않는 결과가 반환되는 경우도 존재한다. 예를 들어 동일한 환경에서 질의어를 'M.H. LEE'과 'J.H. KANG'을 입력한 경우 동명이인의 문제로 7회 반복에 412명의 유사데이터가 검색되었다. 그러나 이러한 문제는 비단 본 논문에서만 발생하는 문제라기보다는 정보검색의 일반적인 문제이므로 본 논문에서 더 이상 논의하지는 않는다. 이러한 환경에서 여러 키워드들을 생성하여 7회 반복을 수행했을 경우, 검출율(recall)은 0.96의 값을 보인 반면 정확도(precision)는 0.71의 값을 나타냈다. 반복된 실험의 결과 평균적으로 5회 이상의 반복이 수행되면 recall의 값은 크게 증가하지 않고 precision의 값은 낮아지는 것을 알 수 있다.

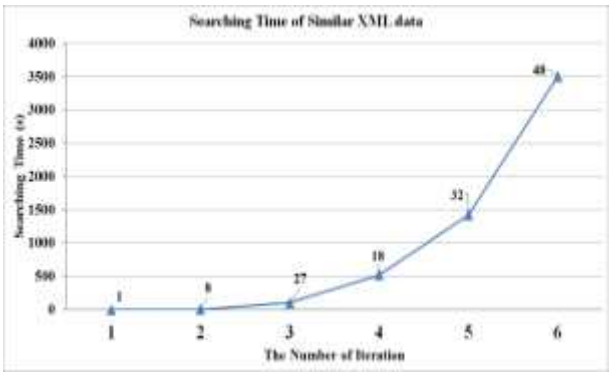

Fig. 9. Searching time according to iteration

평가를 위한 두 번째 방법은 수행 시간의 평가이다. [그림 9]는 약 100M의 sigmod 논문집 데이터를 대상으로 'Gio Wienderhold', 'S.Jerroid Kaplan', 'Daniel Sangalowicz' 세 저자를 질의어로 수행한 검색 시간을 보인다. 검색을 위한 f는 2이다. 그래프에 나타난 데이터의 레이블은 각 반복이 진행되면서 새롭게 검색된 유사 정보의 수이다. 예를 들어 1회 반복의 경우 1명의 저자가 추가로 검색되었으며, 2회 반복의 경우 8명의 또 다른 저자가 검색되었으며, 3회 반복 했을 때 27명의 새로운 저자가 검색되었다. 논문에서 제안한 방법은 새로운 데이터가 검색되면 해당 데이터를 새로운 질의어로 하여 다음 검색의 입력으로 사용한다. 또한 이전 사용된 질의어 역시 새로 검색된 단어들과 함께 새로운 조합으로 구성되어 다음 검색의 입력으로 사용되므로 유사 데이터가 많이 찾아지면 그 수행 시간은 기하급수적으로 증가한다. 그러므로 응용의 성능과 필요에 따라 적절히 반복의 수를 조절이 필요할 것으로 사료된다.

V. Conclusion

본 논문에서는 정보교환의 표준으로 자리 잡은 XML문서로부터 유사 정보를 추출하기 위하여 유사 정보 검색을 위한 효과적인 처리단위를 제안하고 이를 결정하기 위하여 구조기반 접근 방법과 수치기반 접근 방법을 제안하였다. 또한 논문은 제안한 알고리즘을 검증하기 위한 프로토타입 시스템을 구현하여 평가하였다. 본 논문의 결과는 XML 데이터로부터 정보를 검색하기 위한 다양한 응용에서 참조할 수 있을 것이며, 제안한 두 방법은 향후 상호 보완하여 다양한 유사정보 추출을 위해서 사용될 수 있을 것으로 기대한다. XML 데이터로부터 유사정보를 검색하기 위한 기본 아이디어는 HTML 문서로부터 얻어졌다. 그러므로 논문에서 제안하고 있는 알고리즘은 간단한 변경만으로 인터넷상의 주요한 정보형태의 하나인 HTML문서를 위해서도 적용될 수 있다.

현재 논문은 성능평가를 위하여 논문집 XML 문서들을 그 대상으로 하고 있다. 그러나 향후 다양한 응용들에서 사용하는 XML 문서들을 대상으로 실험하고 그 결과를 분석할 것이다.

REFERENCES

[1] Y. Yamada, N. Craswell, T. Nakatoh, and S. Hirokawa, "Testbed for information extraction from deep web," Proc, of WWW 2004, pp. 346-347, 2004.

[2] S. Hirokawa, E. Itoh, and T. Miyahara, "Semi-Automatic Construction of Metadata from a Series of Web

Documents," Proc. Australian Conference on Artificial Intelligence 2003, pp. 942-953, 2003.

[3] T. Miyahara, Y. Suzuki, T. Shoudai, T. Uchida, S. Hirokawa, K. Takahashi, and H. Ueda, "Extraction of Tag Tree Patterns with Contractible Variables from Irregular Semistructured Data," Proc, of PAKDD 2003, pp. 430-436, 2003.

[4] P. T. Nguyen, and H. A. Le, "Finding Similar Artists from the Web of Data: A PageRank Based Semantic Similarity Metric," Proc. of FDSE 2015, pp. 98-108, 2015.

[5] E. Iosif, and A. Potamianos, "Similarity computation using semantic networks created from web-harvested data," Natural Language Engineering Vol.21, No. 1, pp. 49-79, 2015.

[6] H. P. Leung, K. F. Chung and S. C. Chan, "A New Sequential Mining Approach to XML Document Similarity Computation," Proc, of PAKDD 2003, pp. 356-362, 2003.

[7] J. Tekli, R. Chbeir, A. J. M. Traina, C. Traina Jr., and R. Fileto, "Approximate XML structure validation based on document-grammar tree similarity," Information Sciences, Volume 295, pp. 258-302, 2015.

[8] R. Periakaruppan, and R. Nadarajan, "A Prufer Sequence Based Approach to Measure Structural Similarity of XML Documents," Proc. Of OTM Workshops 2013, pp. 639-648, 2013.

[9] C.Y. Wang, X.J. Wu, J. Li, and Y. Ge, "Structural Similarity Evaluation of XML Documents Based on Basic Statistics," Proc. of WISM 2012, pp. 698-705, 2012.

[10] S. Flesca, G. Manco, E. Masciari, L. Pontieri and A. Pugliese, "Detecting Structural Similarities between XML Documents," Proc. of WebDB 2002, pp. 55-60, 2002.

[11] J. Manuel, A. Jimenez, and A. Cuzzocrea, "SemSynX: Flexible Similarity Analysis of XML Data via Semantic and Syntactic Heterogeneity/Homogeneity Detection," Proc. of HAIS 2016, pp. 12-26, 2016.

[12] K.Y. Lee, "A Study on the Development of Ontology based on the Jewelry Brand Information," Journal of the Korea Society of Computer and Information, Vol. 13, No. 7, pp. 247-256, 2008.

[13] S. Akmala, L.H. Shihb, and R. Batres, "Ontology-based similarity for product information retrieval," Computers in Industry, Vol. 65, No. 1, pp.91-107, 2014.

Authors

Jong-Hyun Park is received his Ph.D. and M.S. degrees in computer science from Chungnam National University, South Korea, in 2002 and 2007, respectively, and his B.S. degree in computer science from Woosong University, South Korea, in 1999.

Dr. Park is a visiting professor at Dept. of computer engineering & science in Chungnam National University, South Korea, since 2011. From 2009 to 2010, he has researched at the Research Institute for Information Technology of the Kyushu University in JAPAN. His research interests include recommender system, Context-Awareness, M2M, Ontology, Reasoning, Personalization, Semantic Web, Web Information system, XQuery Processing, XML database, and Database systems.