XML 트리의 노드와 레벨을 사용한 군집화 방법

김우생*

Clustering Technique Using a Node and Level of XML tree

Woosaeng Kim*

이 논문은 2013년도 광운대학교 교내 학술연구비를 지원받았음

요 약

최근 들어 인터넷에서 많이 사용되는 XML 문서들을 효율적으로 접근, 질의, 관리하는 방법들이 연구되고 있다. 본 논문은 XML 문서들을 효율적으로 군집화 하는 새로운 기법을 제안한다. XML 문서의 원소는 대응하는 트리의 노드에 대응하며, 문서에서의 내포 관계는 대응하는 트리의 레벨 관계에 대응한다. 따라서 유사한 XML 문서들은 대응하는 트리들에서 노드의 이름과 레벨이 유사하다. 본 논문에서는 XML 문서의 특징으로 대응하는 트리의 노드 이름과 레벨을 사용하여 군집화를 수행하였다. 제안하는 기법이 좋은 결과를 얻을 수 있음을 실험을 통하여 보였다.

ABSTRACT

Recently, researches are studied in developing efficient techniques for accessing, querying, and managing XML documents which are frequently used in the Internet. In this paper, we propose a new method to cluster XML documents efficiently. An element and an inclusion relationship of a XML document corresponds to a node and a level of the corresponding tree, respectively. Therefore, when two XML documents are similar then their nodes' names and levels of the corresponding trees are also similar. In this paper, we cluster XML documents by using nodes' names and levels of the corresponding tree as a feature of a document. The experiment shows that our proposed method has a good performance.

키워드

XML, XML 군집화, 계층 군집화

Key word

XML, XML Clustering, Hierarchical Clustering

* 정회원 : 광운대학교 컴퓨터과학과 (kwsrain@kw.ac.kr)

접수일자 : 2012. 08. 27
심사완료일자 : 2012. 10. 19

Open Access http://dx.doi.org/10.6109/jkiice.2013.17.3.649

Ⅰ. 서 론

인터넷의 성장은 전 세계에 존재하는 모든 데이터와 정보의 접근을 쉽게 만들면서 많은 데이터들이 다양한 형태의 정보로 생성되는데 이바지하고 있다. 인터넷이 점점 성장하고 발전할수록, 더 많은 정보들은 XML과 같이 구조적으로 풍부한 문서 형태로 존재하게 된다. 웹에서 문서가 많아 질수록, 이와 같이 구조적으로 풍부한 문서들을 자동적으로 검색하고 관리하는 응용들이 요구되고 있다.

XML 문서들에 대한 군집화는 유사한 문서들의 그룹을 만들어 특정한 카타고리 안에서 검색과 처리를 용이하게 하기 위함이다. XML 문서에 대한 적절한 군집화는 체계적인 문서 관리와 문서 저장을 위해서도 효율적이다. 또한 군집화 된 데이터들은 데이터들 간에 일종의 경향 또는 규칙성을 보이고 심지어 주목할 가치가 있는 관련 지식을 보여 주기까지 한다.

본 논문은 같은 DTD에서 생성된 XML 문서들은 유사한 문서들로 간주하여, 이러한 문서들을 효율적으로 군집화 하는 방법을 제안한다. XML문서의 원소들은 내포 구조로 구성되기 때문에 XML은 정렬된 라벨 트리로 모델링 될 수 있다[1]. XML 문서의 원소는 트리의 노드에 대응하며, 문서에서 내포 관계는 트리의 부모와 자식 노드 간의 관계 즉 레벨 관계에 대응한다. 따라서 유사한 XML 문서들은 대응하는 트리들의 노드 이름과 레벨 등이 유사하며, 이를 XML 문서의 특징으로 사용하여 계층 군집화를 시도 하였다. 실험을 통해 제안하는 방법이 군집화를 효율적으로 수행함을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 XML 군집화 관련 연구에 대해 기술하고, 3장은 XML 문서 군집화를 위한 방법 제안한다. 4장은 실제 데이터를 통한 실험을 통하여 제안한 방법이 효율적인지를 조사한다. 마지막으로 5장은 결론을 낸다.

Ⅱ. 관련연구

다양한 XML의 증가로 인해 XML 데이터를 조직하고 군집화 하는 필요성이 증대되고 있다. XML 문서의 구조를 이산 함수로 변환하는 방법이 연구되었다[2]. 이산 함수는 FFT에 의해서 주파수 영역으로 변환된다. FFT의 결과는 x와 y의 값들을 포함하는 복소수 쌍들이며 n 차원 벡터들로 간주되어 유클리디안 거리 척도를 사용하여 비교된다. 문서간의 구조에 대한 공통 구조의 존재 여부를 0과 1로 표현하는 비트를 이용하여 비트맵 인덱스에 기반한 군집화 기법이 제안되었다[3,4]. BitCube는 3 요소 즉, 문서, 경로, 단어의 3 차원 비트맵 인덱스로 표현된다. BitCube 인덱스들은 문서들을 분할해서 군집화하기 위해 bit-wise 거리 척도를 활용한다. XML 데이터를 위한 특징들로 문서로부터 추출한 내용 정보와 태그 경로들로부터 유도되는 구조 정보들을 사용하는 방법이 제안되었다[5].

이 방법은 XML 문서 트리들을 트랜잭션 데이터 즉, 속성들을 가진 객체들로 사상하는 것을 허용하는 XML 표현 모델의 정의 안에서 트리 투플이라는 개념을 소개하며, 군집화 기법이 XML 트랜잭션의 영역 안에서 개발되고 적용되었다. XML 문서의 빈발 경로나 대표 경로 등에 기반한 군집화 기법이 제안 되었으며[6,7]. 다양한 구조를 가지는 XML 문서의 경로 구조를 중심으로 빈발 구조에 대한 유사성 기반의 점진적 군집화 기법도 제안되었다[8].

XML 문서를 구성하는 원소의 순서와 발생 빈도를 동시에 고려할 수 있는 순차 패턴을 이용하여 일정한 지지도를 만족하는 빈발 구조 패턴을 추출하여 유사 구조 문서를 그룹화 하여 주요 항목 기반의 클러스터를 생성하고, 클러스터 할당 이익에 대한 연산을 통해 점진적 군집화를 수행하였다. 유사한 XML 문서들은 대응하는 트리들에서 노드의 이름과 레벨 등이 유사한다. 이러한 성질은 유전 알고리즘의 평가함수로 만들어 군집화를 수행하였다[9].

Ⅲ. XML 문서의 노드 레벨 벡터와 군집화 기법

그림 1의 Club의 DTD에 의해 생성되는 일부 문서들을 대응하는 트리들로 표현하면 그림 2의 (a), (b)로 표현된다. 예를 들어, 그림 2(a)는 트리의 첫 레벨에 있는 노드인 club, 두 번째 레벨에 있는 노드들인 clubname, member 그리고 세 번째 레벨에 있는 노드들인 name, phone, addr로 표현된다.

이처럼 XML 문서를 대응하는 트리 구조의 노드들의 이름과 레벨로 표현할 때, 같은 DTD에 의해 생성되는 유사한 문서들은 같은 레벨의 같은 노드 이름을 많이 공유함을 알 수 있다. 예를 들어, Club DTD에 의해 생성되는 그림 2(a), (b)는 첫 번째 레벨의 club은 루트로 같고, 두 번째 레벨의 clubname과 member, 그리고 세 번째 레벨의 name과 phone이 같다.

```
Club
<!ELEMENT club (clubname, member+)>
<!ELEMENT clubname (#PCDATA)>
<!ELEMENT member (name, phone, addr?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT addr (#PCDATA)>
```

그림 1. Club DTD
Fig. 1 Club DTD

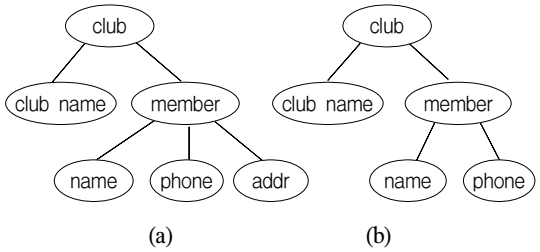

그림 2. Club DTD에 대응하는 트리들
(a) Club1 (b) Club2
Fig. 2 Trees correspond to Club DTD
(a) Club1 (b) Club2

XML 문서를 대응하는 트리 구조의 정보로 표현하기 위해, 본 논문에서는 주어진 모든 XML 문서에 포함된 원소들을 알파벳 순서로 정렬한 후 각 원소에 대응하는 노드의 레벨 값으로 표현한다[1]. 따라서 그림2의 XML 문서들은 표1과 같은 노드 레벨 행렬로 표현된다. 노드 레벨 행렬에서 세로 축은 XML 문서들이고, 가로 축은 XML 문서들에 포함된 모든 원소 이름들의 알파벳 순서이며, 각 셀은 XML 문서에 포함된 원소 이름에 대응하는 노드의 레벨을 나타낸다. 따라서 Club1 노드 레벨 벡터의 첫 번째 셀은 원소 addr에 대응하는 노드의 레벨 값인 3이고 나머지 셀들도 이와 같은 방법으로 표현된다. Club2 는 원소 addr가 없기 때문에 Club2 노드 레벨 벡터의 첫 번째 셀은 '-'로 표현된다. 유사한 문서들의 경우 DTD의 특별한 기호인 *, ?가 붙지 않은 원소에 대응하는 노드의 레벨 값은 항상 같다.

표 1. XML 문서들에 대응하는 노드 레벨 행렬
Table. 1 Node-level matrix corresponding to XML documents

	addr	club	cname	mem	name	pho
Club1	3	1	2	2	3	3
Club2	-	1	2	2	3	3

3.1. DTD 정보가 있을 때의 군집화 기법

XML 문서의 DTD가 있을 경우에는 먼저 주어진 모든 DTD에 대응하는 DTD 노드 레벨 행렬을 만든다. 이 때 DTD에 포함될 수 있는 특수 기호인 ?, +, *의 경우에는 해당 원소가 한 번만 발생하는 것으로 간주한다. 예를 들어서 Actor DTD가 그림 3과 같을 때 Actor와 Club의 DTD에 의해 만들어 지는 DTD 노드 레벨 행렬은 표 2와 같다.

```
Actor
<!ELEMENT actor (name, addr?, phone?, movie)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT addr (#PCDATA)>
<!ELEMENT movie (title, year?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA)>
```

그림 3. Actor DTD
Fig. 3 Actor DTD

표 2. Actor와 Club의 노드 레벨 행렬
Table. 2 Node-level matrix made by Actor and Club

	act	adr	clu	cln	mb	mv	nm	ph	tit	yr
Actor	1	2	-	.	-	2	2	2	3	3
Club	-	3	1	2	2	-	3	3	-	-

1) 만약 한 XML 문서에 같은 원소가 여러 레벨에 걸쳐 존재하면, 레벨 값은 루트에 가장 가까운 레벨의 위치이다.

다음으로 군집화하려는 XML 문서에 대응하는 노드 레벨 벡터를 만든 후, 이 노드 레벨 벡터를 DTD 노드 레벨 행렬의 각 벡터들과 유사한지 조사한다. 이 때 두 벡터의 각각의 셀에 있어서 같은 셀[2]의 수가 많은 경우가 두 벡터간의 유사도가 높은 경우이다. 마지막으로 군집화하고자 하는 문서는 자신의 노드 레벨 벡터와 가장 유사도가 높은 DTD 노드 레벨 행렬의 특정 벡터로 군집화시킨다. 예를 들어, 표2와 같은 DTD 노드 레벨 행렬이 있을 때 Club2는 다음과 같은 방법으로 군집화가 된다. (i) Club2의 노드 레벨 벡터 (-, -, 1, 2, 2, -, 3, 3, -, -)가 만들어진다. (ii) Club2의 노드 레벨 벡터를 표2의 DTD 노드 레벨 행렬의 각 벡터들과 유사한지를 검사한다. (iii) 표2의 Actor 노드 레벨 벡터와의 유사도는 0이지만 Club 노드 레벨 벡터와의 유사도는 5이므로, Club2는 Club 문서로 군집화 된다.

3.2. DTD 정보가 없을 때의 군집화 기법

본 논문은 DTD가 없는 XML 문서들의 군집화를 위해서 계층 군집화 기법을 사용한다. 계층 군집화 기법은 중첩(nested) 군집들의 계층 구조를 생성한다. 만약 R_1 내의 각 군집이 R_2의 진부분 집합이면, k개의 군집을 포함하는 군집 R_1은 r ($<k$)개의 군집을 포함하는 군집 R_2에 중첩된다고 말한다. 계층 군집화 기법은 군집을 형성하는 방향에 따라서 응집형(agglomerative)과 분리형(divisive)으로 구분된다. 모든 패턴의 숫자를 n이라 가정할 때 일반적인 응집형 군집화 기법은 다음과 같다.

① 각 한 개의 패턴으로 구성된 n개의 군집들로부터 시작한다.
② 단계③을 $n-1$ 번 반복한다.
③ 가장 유사한 군집 쌍 C_i 와 C_j 를 하나의 군집으로 합친다. 만약 하나 이상의 쌍이 존재한다면 첫 번째 쌍을 합친다.

그림 4. 응집형 군집화 알고리즘
Fig. 4 Agglomerative clustering algorithm

응집형 군집화 기법에서는 군집 간 유사도를 결정하는 방법에 따라 단일 연결(single-linkage), 완전 연결(complete-linkage) 알고리즘 등이 존재한다. 단일 연결 알고리즘에서는 두 군집 사이의 거리는 각 군집에 있는 패턴 간 거리 중에서 가장 짧은 거리로 정의한다. 만약 Ci와 Cj가 군집이라면, 그들 간 거리 d_{SL}은 식 (1)과 같이 정의된다. 여기서 $d(X,Y)$는 패턴 X와 Y 사이의 거리를 나타낸다.

$$d_{SL}(C_i, C_j) = \min_{X \in C_i,\, Y \in C_i} d(X, Y) \quad (1)$$

반면 완전 연결 알고리즘은 두 군집 사이의 거리를 각 군집에 있는 패턴 간 거리 중에서 가장 긴 거리로 정의한다. 만약 Ci와 Cj가 군집이라면, 그들 간 거리 d_{CL}은 식 (2)와 같이 정의된다. 여기서 $d(X,Y)$는 패턴 X와 Y 사이의 거리를 나타낸다.

$$d_{CL}(C_i, C_j) = \max_{X \in C_i,\, Y \in C_i} d(X, Y) \quad (2)$$

군집화 과정을 위해 중요한 하나의 이슈는 패턴간의 유사도를 어떻게 정량화하는가에 있다. 패턴간의 유사성 (또는 비유사성)을 측정하는 가장 일반적인 척도는 거리, 특별히 유클리디안 거리를 사용한다. 그러나 본 논문에서는 XML문서를 노드 레벨 벡터로 표현하기 때문에, 두 벡터간에 얼마나 같은 셀들이 있는가로 문서간의 유사도를 나타낸다. 따라서 두 XML 문서 X와 Y의 대응하는 문서 노드 레벨 벡터들을 Vx 와 Vy로 표기할 때, 두 문서의 유사도 $f(X, Y)$는 식 (3)과 같이 정의된다.

$$f(X, Y) = Cell\_wise\ AND\ (V_x, V_y) \quad (3)$$

여기서 Cell_wise AND 연산은 두 벡터 Vx, Vy의 각각의 셀에 있어서 같은 셀들의 숫자를 반환한다. 예를 들어, 3개의 Actor 문서와 2개의 Club 문서가 있다고 가정할 때, 5개의 문서들로 만들어 지는 노드 레벨 행렬은 표 3과 같다. 표 3에서 가장 유사한 2개의 군집은 A1과 A2

2) 같은 셀이란 두 셀의 값이 같은 숫자를 가지는 경우로 정의한다.

이다. 왜냐하면 두 군집간의 Cell-wise AND 연산의 결과는 6으로 다른 어떤 2개의 군집간의 Cell-wise AND 연산의 결과보다 크기 때문이다. 따라서 $\{A_1, A_2\}$를 하나의 군집으로 형성한다. 마찬가지 방법으로 다음으로 가장 유사한 2개의 군집은 Cell-wise AND 연산의 결과가 5인 C_1과 C_2이기 때문에 $\{C_1, C_2\}$를 하나의 군집으로 형성한다. 마지막으로 가장 유사한 2개의 군집은 Cell-wise AND 연산의 결과가 4 인 A_1과 A_3 또는 A_2와 A_3이기 때문에 $\{A_1, A_2, A_3\}$를 하나의 군집으로 형성한다. 따라서 계층 군집화 기법을 적용하여 5개의 문서들에 대해 2개의 군집화를 수행하면 $\{A_1, A_2, A_3\}$과 $\{C_1, C_2\}$로 제대로 군집이 이루어짐을 알 수 있다.

표 3. 5개의 XML 문서로 만들어 진 노드 레벨 행렬
Table. 3 Node-level matrix made by 5 XML documents

	act	adr	clu	cln	mb	mv	nm	ph	tit	yr
A1	1	2	-	-	-	2	2	2	3	3
C1	-	3	1	2	2	-	3	3	-	-
A2	1	-	-	-	-	2	2	2	3	3
A3	1	-	-	-	-	2	2	-	3	-
C2	-	-	1	2	2	-	3	3	-	-

Ⅳ. 실험 및 분석

본 연구는 위스콘신 대학의 XML 데이터 뱅크에서 제공하는 데이터들을 사용하여 제안하는 방법의 효율성을 실험하였다[10]. 이 데이터 뱅크는bibliography, club, company profiles, stock quotes, department, personal information, movies, actor와 같은 8개의 DTD를 제공한다. XML 문서들의 군집화를 실험하기 위해 각 DTD에 대하여 10개씩의 문서들을 생성하여 모두 80개의 문서들을 사용하였다.

우선 DTD의 정보가 있는 XML 문서들을 군집화하기 위하여 모든 DTD에 대응하는 DTD 노드 레벨 행렬과 각 문서에 대응하는 노드 레벨 벡터를 만들어 사용하였다. 그림 5는 80개의 문서들에 대한 군집화의 결과이다. 각 DTD에 대하여 제대로 군집화 된 문서들의 수(NCC)와 잘못 군집화 된 문서들의 수(NIC)를 계산하였다. 그림에서 보는 것처럼 모든 문서들이 제대로 군집화가 되는 것을 알 수 있다. 이것은 당연한 결과로서 서로 다른 DTD들에 의해 생성되는 문서들간에 노드의 이름과 레벨이 같은 경우가 일부 있을 수 있으나 대부분은 서로 다르기 때문에, 주어진 문서의 노드 레벨 벡터는 대응하는 DTD 노드 레벨 벡터와 가장 큰 Cell-wise AND 연산 값을 갖기 때문이다.

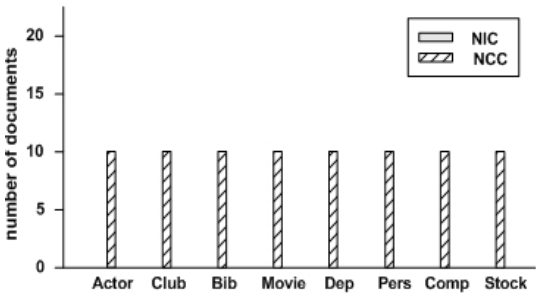

그림 5. 군집화된 문서들의 숫자
Fig. 5 Number of clustering documents

다음으로 DTD의 정보가 없는 XML 문서들을 군집화하기 위하여 각 문서에 대응하는 노드 레벨 벡터를 사용하였다. 그림 6은 XML 문서들에 대해 계층 군집화의 단일 연결 알고리즘을 적용한 성장 그래프이다. 그림에서 가로축은 문서이며 세로축은 유사도 척도이다.

각 문서는 이해를 돕기 위하여 다음의 번호로 표기하였다: Actor(1～5), Club(6～10), Bibliography(11～15), Movie(16～20), Department(21～25), Personal(26～30), Company(31～35), Stock(36～40). 그림에서 보듯이 모든 문서들은 제대로 군집화가 되었음을 알 수가 있다. 그림에서 Department 문서들(21～25)이 가장 먼저 군집화가 되는 것을 알 수 있다. 이것은 Department DTD가 원소를 가장 많이 갖고 있기 때문에 해당 문서들의 노드 레벨 벡터들이 가장 많은 숫자 셀들을 포함하고 있기 때문이다. 그림 7은 같은 문서들에 대해 계층 군집화의 완전 연결 알고리즘을 적용한 성장 그래프이다. 완전 연결 알고리즘의 경우 군집화 되는 순서는 그림 7과는 약간 다르나 제대로 군집화가 되는 것을 알 수 있다.

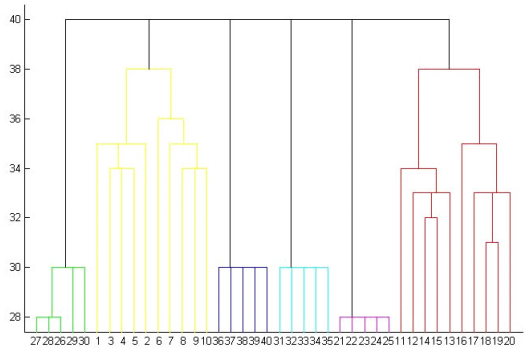

그림 6. 단일연결 알고리즘에 의한 성장 그래프
Fig. 6 Dendrogram by single-linkage algorithm

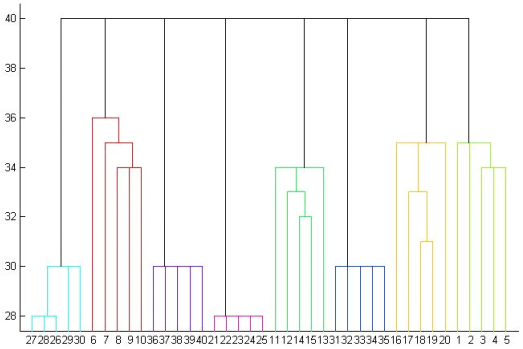

그림 7. 완전연결 알고리즘에 의한 생성 그래프
Fig. 7 Dendrogram by complete-linkage algorithm

V. 결 론

같은 DTD에서 생성된 XML 문서들은 유사하며, 대응하는 트리들에서의 노드 이름과 레벨도 유사하다. 본 논문은 이러한 XML 문서들을 군집화하기 위해 문서의 특징으로 노드 레벨 벡터를 제안하였다. 본 논문은 DTD 정보가 있는 XML 문서들의 경우에는 DTD 노드 레벨 행렬과 문서 노드 레벨 벡터의 유사도를 조사하여 군집화를 하였다. 반면 DTD 정보가 없는 XML 문서들의 경우에는 계층 군집화 기법에 문서 노드 레벨 벡터를 적용하여 군집화를 수행하였다. 실험 결과 본 논문에서 제안하는 방법은 군집화를 제대로 형성하며 효율적으로 수행하였음을 보여 준다.

참고문헌

[1] R.Behrens, “A Grammar based model for XML schema integration,” *Proc. of the 17th British National Conf. on Databases*, pp.172-190, 2000.

[2] H.Lee, “An Unsupervised clustering technique of XML documents based on function transform and FFT,” *Journal of Korea Information Processing Society*, 2007.

[3] J.Yoon, V.Raghavan, V.Chakilam, “BitCube: clustering and statistical analysis for XML documents,” *Proc. of the 13th Int. Conf. on Scientific and Statistical Database Management*, Fairfax, Virginia, 2001.

[4] J.Yoon, V.Raghavan, V.Chakilam, L.Kerschberg, “BitCube: a 3-D bitmap indexing for XML documents,” *Journal of Intelligent Information Systems*, Vol. 17, pp.241-254, 2001.

[5] A.Tagarelli, A.Greco, “Toward semantic XML clustering,” *6th SIAM International Conference on Data Mining*, pp. 188-199. Bethesda, Maryland, USA, 2006.

[6] 이정원, 이기호, “유사성 기반 XML 문서 분석 기법”, 정보과학회논문지: 소프트웨어 및 응용 제 29권 제 5-6호, 2002.6.

[7] 황정희, 류근호, “XML 문서의 공통 구조를 이용한 클러스터링 기법”, 정보과학회논문지 D-데이타베이스 제 32권 제 6호, 2005.12.

[8] 황정희, 류근호 “유사 구조 기반 XML 문서의 점진적 클러스터링,” 정보과학회 논문지- 데이터베이스 제 31권 제 6호, 2004. 12.

[9] 김우생, “유전자 알고리즘을 통한 XML 군집화 방법”, 대한전자공학회 논문지, 2012.5.

[10] Niagara Query Engine, http://www.cs.wisc.edu/niagara/data.html

저자소개

김우생(Woosaeng Kim)

1985년 서울대 수료 및 텍사스주립 대학 전산학과 졸업 (학사)
1987년 미네소타 주립대학 전산학과 졸업(석사)
1991년 미네소타 주립대학 전산학과 졸업(박사)
1987년-1988년 현대전자. 제우스 컴퓨터 과장
1992년 - 현재: 광운대학교 컴퓨터과학과 교수
※관심분야 : 데이터베이스, 멀티미디어