Journal of Korea Multimedia Society Vol. 7, No. 10, October 2004(pp. 1354-1369)

웹 문서로부터 논리적 구조 추출

이민형[†], 이경호[††]

요 약

본 논문에서는 웹 문서를 XML 문서로 변환하기 위한 논리적 구조분석 방법을 제안한다. 제안된 방법은 비주얼 그룹화, 요소 식별, 그리고 논리적 그룹화의 세 단계로 구성된다. 특히 정교한 수준의 논리적 구조분석을 지원하기 위하여 특정 주제에 속하는 문서 유형의 논리적 계층 구조를 효과적으로 기술할 수 있는 문서 모델을 정의한다. 제안된 방법은 비주얼 그룹화를 통해서 추출된 시각적 계층구조와 문서 유형에 대한 논리적 구조 정보를 기술한 문서 모델에 기반하기 때문에 보다 정교한 수준의 구조 분석을 지원한다. 제안된 방법의 성능을 평가하기 위하여 웹으로부터 추출한 다수의 HTML 문서를 대상으로 실험한 결과, 기존 연구와 비교하여 논리적 구조분석을 성공적으로 수행하였다. 제안된 방법은 논리적 구조분석의 최종 결과로서 XML 문서를 생성하기 때문에 문서의 재 사용성을 높인다.

Extracting Logical Structure from Web Documents

Min-Hyung Lee[†], Kyong-Ho Lee[††]

ABSTRACT

This paper presents a logical structure analysis method which transforms Web documents into XML ones. The proposed method consists of three phases: visual grouping, element identification, and logical grouping. To produce a logical structure more accurately, the proposed method defines a document model that is able to describe logical structure information of topic-specific document class. Since the proposed method is based on a visual structure from the visual grouping phase as well as a document model that describes logical structure information of a document type, it supports sophisticated structure analysis. Experimental results with HTML documents from the Web show that the method has performed logical structure analysis successfully, compared with previous works. Particularly, the method generates XML documents as the result of structure analysis, so that it enhances the reusability of documents.

Key words: Web Document Analysis(웹 문서 분석), Logical Structure(논리적 구조), XML, Information Extraction(정보 추출)

1. 서 론

웹 문서는 주로 HTML (Hypertext Markup Language)[1]로 기술되어 있다. HTML은 데이터를 사용자에게 전달하기 위한, 즉 문서의 내용을 시각적으로 랜더링하기 위한 웹 문서 기술 표준이다. 따라서 컴퓨터로 하여금 정보를 처리 및 가공하게 한다는 측면에서는 한계를 갖는다.

한편, XML (eXtensible Markup Language)[2]은 논리적 구조를 표현할 수 있다는 장점 때문에 차세대 웹 문서 표준으로 그 중요성이 널리 인식되고 있다. 특히, 웹의 이질적인 정보를 논리적으로 조직하고 분류할 수 있는 방법이 정보화 지식 사회의 주요한 문제로 떠오르면서 XML이 이에 대한 해결책으로 대두되고 있다. 따라서, HTML 문서로부터 유용한 정보를 추출하여 XML 문서로 변환하는 방법이 요구된다.

※ 교신저자(Corresponding Author) : 이민형, 주소 : 서울시 서대문구 신촌동 134(120-749), 전화 : 02)2123-3878, FAX : 02)365-2579, E-mail : mhlee@icl.yonsei.ac.kr
접수일 : 2004년 2월 11일, 완료일 : 2004년 5월 4일
[†] 준회원, 연세대학교 대학원 컴퓨터과학과 석사과정
[††] 정회원, 연세대학교 컴퓨터산업공학부 조교수
(E-mail : khlee@cs.yonsei.ac.kr)

일반적으로, 인간은 문서로부터 텍스트 영역의 기하적 특성이나 어휘 정보를 이용하여 제목 또는 요약 등의 논리적 구성 요소를 식별하고, 이를 병합하여 절 구조와 같은 복합적인 구성 요소를 식별함으로써 문서의 논리적 계층 구조를 인식한다. 이와 같이 텍스트 영역의 기하적 및 어휘적 특성으로부터 직접적인 식별이 가능한 논리적 요소를 주 구조(primary structure)라고 하며 이미 식별된 다수의 구성 요소들을 병합함으로써 추출 가능한 구성 요소를 부 구조(secondary structure)라고 한다[3]. 따라서, 웹 문서로부터 XML 문서를 생성하기 위해서는 주 구조는 물론이고 부 구조에 대한 논리적 구조분석이 이루어져야 한다.

그러나, HTML 문서의 논리적 구조분석에 관한 기존 연구의 대부분은 주 구조에 해당하는 구성 요소만을 추출하거나 단순한 수준의 구조분석을 지원한다. 한편, 웹 문서로부터 논리적인 계층 구조의 효과적인 추출을 위해서는 문서 유형의 논리적 계층 구조에 대한 다양한 정보를 표현할 수 있는 문서 모델이 요구된다. 기존 연구의 대부분은 단순한 수준의 문서 모델을 제공한다.

본 논문에서는 웹 문서로부터 논리적인 구조 정보를 추출할 수 있는 효율적인 방법을 제안한다. 제안된 방법은 비주얼 그룹화(visual grouping), 요소 식별(element identification), 그리고 논리적 그룹화(logical grouping)의 세 단계로 구성된다. 특히, 제안된 방법은 정교한 수준의 구조분석을 위하여 문서 모델을 효율적으로 표현할 수 있는 언어인 MEDL (multi-level element description language)을 제안한다. MEDL은 논리적 계층 구조를 기술하기 위하여 특정 주제에 속하는 문서 집합이 포함할 수 있는 논리적 구성 요소의 종류, 포함관계, 그리고 빈도수 등에 대한 다양한 정보를 기술한다.

제안된 방법의 성능을 평가하기 위하여 웹으로부터 추출한 HTML 문서 집합을 대상으로 실험한 결과, 기존 연구와 비교하여 논리적인 구조분석을 성공적으로 수행하였다. 특히 제안된 방법은 논리적인 구조분석의 최종 결과로서 XML 문서를 생성하기 때문에 문서의 재 사용성을 지원한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 통하여 웹 문서의 논리적 구조분석 방법에 대한 기존의 연구 결과를 간략히 기술한다. 3절에서는 문서 모델을 기술하기 위하여 정의된 MEDL을 자세히 설명한다. 4절에서는 제안된 구조 분석 방법을 비주얼 그룹화, 요소 식별, 그리고 논리적 그룹화의 세 단계로 구분한 후 각각에 대한 자세한 설명을 기술한다. 5절에서는 실험 결과를 통하여 제안된 방법의 성능을 기존 연구와 비교 및 분석한다. 마지막으로 6절에서는 결론 및 향후 연구 방향을 기술한다.

2. 관련 연구

일반적으로 웹 문서의 논리적 구조분석에 관한 기존 연구는 표현 범위와 방법은 다르지만 문서 유형에 대한 지식을 표현하기 위하여 문서 모델을 사용한다. 본 절에서는 표 1과 표 2와 같이 추출 가능한 구조의 종류, 문서 모델의 표현 범위, 그리고 XML 문서의 자동 생성 여부의 세 가지 측면에서 각각의 특징과 문제점을 간략히 기술한다.

Lim과 Ng[4]는 HTML 문서를 XML 문서로 자동 변환하는 Html2Xml을 제안한다. Html2Xml은 XML 변환 시 구조에 영향을 미치는 정도에 따라 HTML 태그에 우선순위를 부여한 후, 태그의 우선순위에 기반하여 문서의 계층 구조를 생성한다. 따라서, 태그의 우선순위만을 가지고 구조를 분석하기 때문에 우선순위가 정확하지 않으면 잘못된 구조의 XML 문서를 생성할 수 있다. 또한, 단순히 텍스트 자체를 요소 이름으로 사용하는 제약을 갖는다. 한편, 김승원 등[8] 은 Html2Xml의 단점을 보완하는 방법을 제안한다. 이를 위해서 태그의 우선순위 설정 및 XML 태깅

표 1. 성능 평가 기준

기준 / 기호	추출 가능한 구조의 종류	문서 모델의 표현 범위
△	주 구조	구체적인 표현 기법 없이 단순한 규칙 또는 일반적인 특성
□	제한된 수준의 계층 구조	논리적 구성 요소의 특성을 기술하며 제한된 수준의 논리적 계층 구조를 기술
○	논리적 계층 구조	구성 요소 사이의 관계 및 빈도수 등을 포함한 논리적 계층 구조 기술

표 2. 웹 문서의 논리적 구조분석 방법

저자	연도	특 징	구조 종류	문서모델 표현범위	XML 자동생성
Lim 과 Ng[4]	2000	HTML 문서를 자동으로 XML 문서로 변환하는 Html2Xml를 제안한다. HTML 태그의 우선순위에 따라 문서의 계층구조를 분석한다.	△	△	○
Taniar 등[5]	2000	추출된 HTML 문서의 스키마에 수동 방식의 맵핑 도구를 적용하여 XML 문서를 생성한다.	□	□	×
Chung 등[6]	2001	특정 주제 HTML 문서를 대상으로 키워드와 베이즈 분류기에 기반하여 요소를 식별하는 Quixote를 제안한다.	□	□	○
Han 등[7]	2001	다섯 개의 휴리스틱한 방법으로 구성된 WRAP Elite를 제안한다. 특히 태그의 우선순위에 따라 객체를 추출한다.	△	△	○
김승원 등[8]	2002	Html2Xml에 사용자와의 상호작용을 적용하여 XML 구조 분석의 성능을 향상시킨다.	□	△	×
오금용 등[9]	2002	공통 패턴을 갖는 HTML 문서집합에 특정 DTD를 적용하여 요소를 추출한다.	□	□	○

을 사용자와의 상호작용을 통해서 수행한다.

Taniar 등[5]은 효율적인 웹 검색을 목적으로 스키마 추출 및 맵핑(mapping) 도구를 적용하여 평면적인 HTML 문서를 XML 문서로 변환한다. 따라서, 문서를 변환할 때 마다 맵핑 과정에 사용자가 개입하여 문서를 변환하여야 한다.

Chung 등[6]은 특정 주제의 웹 문서로부터 논리적 구조를 추출하기 위한 Quixote를 제안한다. Quixote는 논리적 구조분석을 위해서 키워드 정보 또는 베이즈 분류기(Bayes classifier)를 적용하여 요소를 식별한다. 또한 식별된 요소, HTML 태그, 그리고 태그의 반복 정보를 이용하여 계층 구조를 갖는 XML 문서를 생성한다. Quixote는 HTML 문서의 텍스트만을 가지고 요소를 식별하기 때문에 구조적으로 부적합한 요소를 식별할 수 있다.

Han 등[7]은 HTML 문서로부터 구조 정보를 추출하기 위한 래퍼(wrapper) 프로그램의 반자동 생성을 위한 시스템인 XWRAP Elite를 제안한다. 일반적으로 래퍼는 문서의 논리적 구조분석이라기보다는 HTML 문서로부터 의미 있는 부분을 추출하는데 초점이 맞추어져 있기 때문에 본 논문과는 목적이 다르다고 볼 수 있다.

오금용 등[9]은 문서의 유사성에 기반하여 유사한 패턴을 갖는 문서 집합을 형성한다. 이렇게 형성된 문서 집합으로 부터 특정 DTD(Document Type Definition)[2]에 정의된 요소를 추출하여 XML 문서를 생성한다.

한편 XWRAP Elite와 같은 기존 래퍼 프로그램들은 문서의 논리적 분석보다는 필요한 정보를 추출하는데 초점을 맞추고 있다. 또한 기존 래퍼 프로그램들은 유사한 패턴을 가진 문서나 동일한 방식으로 제작된 문서 그룹을 대상으로 한다. 또한 래퍼 프로그램은 이러한 문서에 대해 패턴을 분석 또는 예제 입력 파일을 통해 구조를 분석하여 정보를 추출해 낸다[7,10,11]. 반면 제안된 방법은 공통적인 구조를 갖는 문서 집합을 대상으로 하지 않고 동일한 주제를 갖는 문서를 대상으로 하기 때문에 그 목적이 다르다고 할 수 있다. 예를 들어, 기존 래퍼 프로그램들은 공통된 주제의 문서라도 서로 다른 구조를 갖는 100개의 문서가 존재한다면 100개의 예제 입력 또는 패턴이 필요하지만 제안된 방법은 공통된 주제에 대한 하나의 문서 모델만이 존재하면 논리적 구조를 추출해 낼 수 있다.

3. 문서 모델

본 절에서는 특정 주제에 속하는 문서 집합에 대한 문서 모델을 기술할 수 있는 언어인 MEDL을 자세히 기술한다. MEDL은 논리적 구성 요소가 포함할 수 있는 요소의 종류, 순서, 그리고 반복 횟수 등에 대한 정보를 정규 수식으로 표현한다. 특히 주 구조에 대하여 이를 식별하는데 필요한 키워드 및 문자열 패턴과 같은 어휘 정보를 기술한다.

한편, XML은 문서의 논리적 구조 정보를 DTD로 기술한다. 제안된 방법은 논리적 구조 정보와 더불어 주 구조의 어휘적 특성을 기술하기 위하여 그림 1과 같이 XML DTD의 엘리먼트(element) 선언 방법을 확장하여 MEDL을 정의한다. 그림 2는 제안된 MEDL

```
ElementDeclaration ::= '<!ELEMENT' S ElementType S ContentModel S? '>'
ElementType ::= GenericIdentifier
ContentModel ::= ModelGroup
ModelGroup ::= '(' S? ContentToken (S? Connector S? ContentToken)* S? ')' OccurrenceIndicator?
ContentToken ::= ElementToken|ModelGroup
ElementToken ::= GenericIdentifier OccurrenceIndicator?
GenericIdentifier ::= Name
Name ::= NameStartChar (NameChar)*
NameStartChar ::= 'A'|...| 'Z'| 'a' |...| 'z'
NameChar ::= NameStartChar|Digit| '.' | '-' | '_' | ':'
Digit ::= '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'
Connector ::= ','|'|'|'&'
OccurrenceIndicator ::= '?'|'+'|'*'
NmToken ::= (NameChar)+
NmTokens ::= NmToken (S NmToken)*
S ::= (#x20 | #x9 | #xD | #xA)+
KeywardDeclaration ::= '<!KEYWARD' S ElementType S KeywardTokens? S KeywardPattern? S? '>'
KeywardTokens ::= 'KEYTOKENS' S '"' NmTokens '"'
KeywardPattern ::= 'KEYPATTERN' S '"' regular expression '"'
```

그림 1. MEDL의 E-BNF 형식

에 따라 기술된 이력서(resume) 용 HTML 문서에 대한 문서 모델의 예이다.

4. 논리적 구조분석 방법

본 논문은 HTML 문서로부터 논리적 구조 정보를 추출하여 XML 문서를 생성하는 것을 목적으로 한다. 제안된 방법은 그림 3과 같이 전처리, 논리적 구조분석, 그리고 후처리의 세 단계로 구성된다. 제안된 방법은 먼저 전처리 과정을 통하여 입력으로 주어진 HTML 문서가 XML의 적격성(well-formedness) 요건을 만족하도록 불필요한 태그를 삭제하고 생략된 태그를 추가한다.

두 번째, 논리적 구조분석은 비주얼 그룹화, 요소 식별, 그리고 논리적 그룹화의 세 단계로 구성된다. 비주얼 그룹화는 HTML 문서의 기하적 특성에 기반하여 텍스트 영역의 계층적 그룹을 형성한다. 본 논문에서는 비주얼 그룹화의 결과로 생성된 트리 구조를 비주얼 그룹 트리 (visual group tree) 라고 정의한다. 요소 식별 단계에서는 비주얼 그룹 트리에 문서 모델을 적용하여 논리적 구성 요소를 식별한다. 논리적 그룹화 단계에서는 식별된 요소를 포함하는 비주얼 그룹 트리를 대상으로 요소의 반복, 계층 구조 정보 등을 이용하여 논리적 계층 구조를 추출한다. 추출된 계층 구조를 본 논문에서는 논리 구조 트리 (logical structure tree)라고 정의한다.

마지막으로, 후처리 단계에서는 논리적 구조분석 단계에서 추출된 논리 구조 트리를 문서 모델과 비교하여 오류를 수정한다. 또한 논리 구조 트리를 깊이 우선 탐색(depth first traversal) 하면서 논리적 구조 분석의 최종 결과로서 XML 문서를 생성한다. 각 단계에 대한 자세한 설명은 다음과 같다.

4.1 전처리

제안된 방법은 HTML 문서를 구조화하기 위하여 DOM(Document Object Model)[12] 트리에 기반한다. DOM은 HTML 문서를 구성하는 태그, 속성(attribute), 그리고 텍스트 정보를 노드로 트리를 구성한다. 그림 4는 HTML 문서와 이를 DOM 트리로 표현한 것이다.

한편, HTML 문서로부터 DOM 트리를 구성하기 위해서는 HTML 문서가 XML 적격성 요건을 만족하여야 한다. 예를 들어, 그림 5(a)와 같이 끝 태그가 생략되어 적격성 요건을 만족하지 않는 HTML 문서의 경우, 해당 노드가 포함하는 범위가 불분명하기 때문에 DOM 트리를 바르게 생성할 수 없다. 제안된 방법은 HTML Tidy[13]를 적용하여 그림 5(a)의 HTML 문서를 그림 5(b)와 같이 XML의 적격성 요건을 만족하는 형태로 변환한다.

또한, HTML 문서는 논리적 구조분석에 영향을 미치지 않는 태그를 포함할 수 있다. 전처리 과정에서 텍스트 노드사이에 위치하는 인라인 태그인

```
<!ELEMENT Resume(Education|Employment|Honors|Experience|Skill||Coursework|Objective|Activities|Service)*>
<!ELEMENT Education (Degree|Thesis|Advisor|Organization|Nation|City|State|Date|Positon|Department|Gpa|Honorname)*>
<!ELEMENT Employment (Position|Organization|Jobtype|Nation|Date|City|State|Department)*>
<!ELEMENT Honors (HonorName|Organization|Position)*>
<!ELEMENT Experience (Position|Organization|Jobtype|Nation|Date|City|State|Department|Tool|Os|Programming)*>
<!ELEMENT Skill (Tool|OS|Programming)*>
<!ELEMENT Courswork (Coures)*>
<!ELEMENT Activities (Organization|Sports|Nation|City|Department)*>
<!ELEMENT Services (Organization|Jobtype|Service)*>
<!ELEMENT Degree (Thesis|Advisor|Organization|Nation|City|State|Date|Department|Gpa)*>
<!ELEMENT Organization (Degree|Thesis|Advisor|Nation|City|State|Date|Department|Gpa|Jobtype|HonorName|Position|Sports|Service)*>
<!ELEMENT Jobtype (Organization|Nation|City|State|Date|Department|Tool|Os|Programming)*>
<!ELEMENT Date (Degree|Thesis|Advisor|Organization|Nation|City|State|Date|Department|Gpa|Jobtype)*>
<!ELEMENT HonorName (Organization|Position)*>
<!KEYWARD Education KEYTOKENS "education educational">
<!KEYWARD Degree KEYTOKENS "ph.d. m.s. b.s. b.a. ph.d m.s b.s b.a m.sc. b.sc.bs phd degree Bachelor Bach. Master B.tech msee bsee">
<!KEYWARD Organization KEYTOKENS "institute university society college association seminary ministries school center centre station
univ. labortoty Corp. Universit STMicroelectronis 3M Laboratory Inc hospital Ltd church corp co. campus universidad industries">
<!KEYWARD JobType KEYTOKENS "professor prof. assistance officer director teacher researcher engineer lecturer scientist programmer
instructor manager coodinator tutor chemist associate specialist developer consultant volunteer assistantship internship">
<!KEYWARD HonorName KEYTOKENS "award grant prize nominated scholarship fellowship Dean's merit">
<!KEYWARD Advisor KEYTOKENS "advisor">
<!KEYWARD Experience KEYTOKENS "experience">
<!KEYWARD Skills KEYTOKENS "skills computing Languages">
<!KEYWARD Thesis KEYTOKENS "thesis dissertation">
<!KEYWARD Position KEYTOKENS "member president editor position chair">
<!KEYWARD Language KEYTOKENS "English Language french">
<!KEYWARD EMAIL KEYTOKENS "Email @ email:">
<!KEYWARD Assessment KEYTOKENS "assessment">
<!KEYWARD Activities KEYTOKENS "activities">
<!KEYWARD Interest KEYTOKENS "interest hobby">
<!KEYWARD Honors KEYTOKENS "honors awards HONOURS">
<!KEYWARD Affilations KEYTOKENS "affilations">
<!KEYWARD Publications KEYTOKENS "publication publications books">
<!KEYWARD Employment KEYTOKENS "employment">
<!KEYWARD Gpa KEYTOKENS "gpa g.p.a">
<!KEYWARD Contact KEYTOKENS "contact">
<!KEYWARD Services KEYTOKENS "service">
<!KEYWARD Summary KEYTOKENS "summary">
<!KEYWARD Objective KEYTOKENS "objective">
<!KEYWARD Participation KEYTOKENS "participation seminar convention">
<!KEYWARD Nation KEYTOKENS "uk u.s.a. usa nz australia u.s.a india. india. India vietnam turkey israel france kingdom usa. pakistan.
spain Maxico ireland">
<!KEYWARD City KEYTOKENS "baltimore hamilton clayton boston raleigh cambridge francisco bombay madras panama atlanta istanbul
jerusalem georgia karachi mankapur St. dublin Stanford">
<!KEYWARD State KEYTOKENS "MA.NC.NC MA CA massachusetts California Illinois MN">
<!KEYWARD Coursework KEYTOKENS "Courses course coursework courseworks">
<!KEYWARD Presentations KEYTOKENS "Presentations">
<!KEYWARD Tool KEYTOKENS "word chemdraw macwrite quattro-pro molecular excel powerpoing photoshop premier sas maple lindo
cplex ampl mathematica msoffice tex latex vax/vms autocad mathcad spreadsheet simulink telnet ftp xfractint. netscape explorer simnon
khorous">
<!KEYWARD Os KEYTOKENS "unix windows mac-os MS-dos dos macintosh windows. x-windows ms-windows linux solaris NT hardware
macintosh.">
<!KEYWARD Programming KEYTOKENS "html C/C++ C++ java fortran pascal java PROGRAMMING c++ lisp scheme verilog perl assembly
sql cobol sparc til-shell C/UNIX">
<!KEYWARD Course KEYTOKENS "microprocessors architecture systemdesign chemistry spectroscopy synthesis physiology biology
biochemistry">
<!KEYWARD Sports KEYTOKENS "softball table-tennis tennis climbing">
<!KEYWARD Department KEYTOKENS "department dept. ">
```

그림 2. 이력서 용 HTML 문서를 위한 문서 모델의 예.

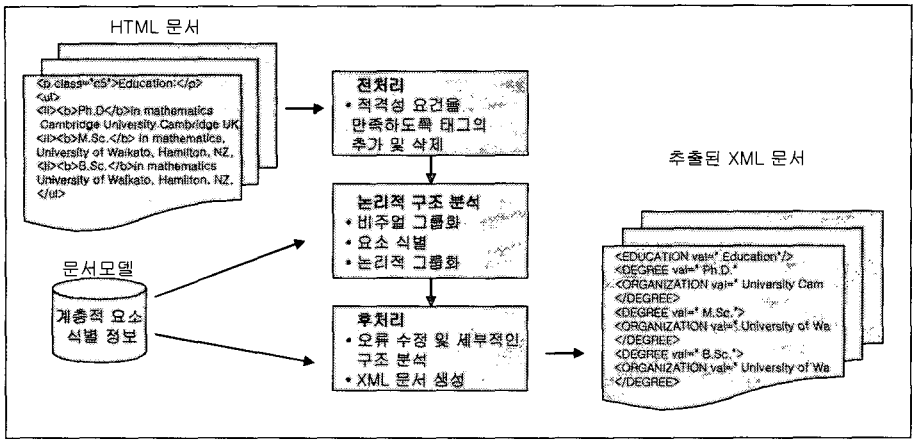
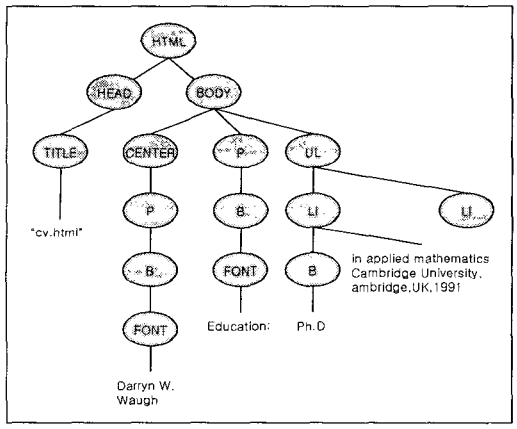


그림 3. 제안된 구조분석 과정.

```
<HTML>
<HEAD><TITLE>cv.html</TITLE>
</HEAD>
<BODY bgColor=#ffffff>
<CENTER>
<P align=center><B>
<FONT size=5>Darryn W. Waugh</FONT>
</B></P></CENTER>
<P><B>
<FONT size=4>Education:</FONT>
</B></P>
<UL>
  <LI><B>Ph.D</B>. in applied mathematics,
  Cambridge University, Cambridge, UK,1991
  <LI><B>M.Sc.</B> in mathematics, University of
  <LI><B>B.Sc.</B> in mathematics &amp; physics,
  Hamilton, NZ, 1985 </LI></UL>
```

(a) HTML 문서의 예



(b) (a)를 DOM 트리로 표현한 결과

그림 4. HTML 문서와 DOM 트리

```
<LI><B>Associate Professor</B>, Johns Hopkins University, Baltimore, U.S.A, 2001 to present
<LI><B>Assistant Professor</B>, Johns Hopkins University, Baltimore, U.S.A, 1998 to 2000
<LI><B>Research Scientist</B>, Cooperative Research Centre for Southern Hemisphere Meteorology,
Monash University, Clayton, VIC, Australia, 1995 to 1997
```

(a) 끝 태그 </LI>가 생략된 원본 HTML 문서의 일부

```
<LI><B>Associate Professor</B>, Johns Hopkins University, Baltimore, U.S.A, 2001 to present</Li>
<LI><B>Assistant Professor</B>, Johns Hopkins University, Baltimore, U.S.A, 1998 to 2000</Li>
<LI><B>Research Scientist</B>, Cooperative Research Centre for Southern Hemisphere Meteorology,
Monash University, Clayton, VIC, Australia, 1995 to 1997 </Li>
```

(b) (a)에 생략된 끝 태그를 삽입한 결과

그림 5. HTML 문서에 Tidy를 적용한 결과

<B>, <I> 등의 태그를 구조에 영향을 끼치지 않는 태그로 간주하여 제거한다.

4.2 논리적 구조분석

논리적 구조분석 방법은 비주얼 그룹화, 요소 식

별, 그리고 논리적 그룹화의 세 단계로 구성된다. 비주얼 그룹화 과정에서는 시각적으로 단락이나 영역을 나누기 위해 사용되었던 HTML 태그들을 이용하여 HTML 문서로부터 개략적인 계층 구조를 추출한다. 요소 식별 단계에서는 비주얼 그룹화에서 생성된 비주얼 그룹 트리에 문서 모델을 적용하여 논리적 구성 요소를 식별한다. 논리적 그룹화에서는 식별된 요소의 반복 정보 등을 활용하여 논리적 계층 구조를 추출한다.

4.2.1 비주얼 그룹화 (Visual Grouping)

일반적으로, HTML 태그는 논리적 구조를 나타내기 보다는 시각적 표현을 위해 사용된다. 특히, HTML 태그 중에는 일반적으로 절 또는 독립적인 영역을 시각적으로 구분하는데 사용되는 태그들이 존재한다. 또한, 사용자마다 서로 다른 태그를 사용할 수 있지만 일반적으로 동일한 문서 내에서는 일관되게 사용된다. 제안된 비주얼 그룹화는 문서의 영역을 구분하는데 사용되는 HTML 태그 집합에 기반하여 HTML DOM 트리를 계층 구조를 갖는 비주얼 그룹 트리로 재구성한다.

제안된 방법은 비주얼 그룹화를 위하여 두 가지 종류의 태그 집합에 기반한다. 먼저 흔히 절 제목을 나타내기 위하여 사용되는 헤딩 태그 (heading tag)인 태그 <H*n*>을 이용한다. 본 논문에서는 <H*n*> 태그와 같이 해당 태그만으로도 비주얼 그룹의 식별이 가능한 태그를 단일 비주얼 태그 (single visual tag)라고 정의한다.

한편, 문서의 작성자에 따라 절 또는 독립적인 영역의 제목을 표현하기 위하여 복수의 태그를 결합하여 사용할 수 있다. 예를 들어, 단락을 구분하기 위하여 사용되는 <P> 태그와 글자의 크기 정보를 나타내는 <FONT> 태그의 "size" 속성을 결합하여 절 제목을 구분할 수 있다. 이와 같이 절 제목을 나타내기 위하여 사용되는 다수의 태그 집합을 복합 비주얼 태그(complex visual tag)라고 정의한다. 본 논문에서 사용하는 비주얼 태그 및 해당 태그의 우선순위는 표 3과 같다.

제안된 방법은 DOM 트리에 하향식 너비 우선 탐색(breadth first search) 과정을 적용하면서 비주얼 태그를 찾는다. 비주얼 태그 사이에는 우선순위가 존재한다. 예를 들어, 태그 <H2>가 태그 <H3>보다 우선순위가 높으며 먼저 그룹화 된다. 복합 비주얼 태그의 경우, 글자의 크기 정보를 나타내는 "size" 속성을 포함할 경우, 이와 동일한 글자 크기를 갖는 헤딩 태그의 우선순위를 갖는다. 비주얼 태그에 의하여 구분되는 영역을 <GROUP> 태그로 둘러싸며, 특히 비주얼 태그가 둘러싸고 있는 텍스트를 <GROUP>의 "title" 속성 값으로 할당한다. 제안된 비주얼 그룹화 알고리즘에 대한 자세한 기술은 그림 6과 같다.

예를 들어, 그림 7(a)의 경우, 복합 비주얼 태그인 <P><B><FONT size="4">가 반복적으로 사용되었다. 따라서, 제안된 알고리즘은 그림 7(b)와 같이 반복되는 비주얼 태그를 기준으로 비주얼 그룹을 생성한다.

표 3. 비주얼 태그

우선순위	단일 비주얼 태그	복합 비주얼 태그
1	<H1>	<P><B><FONT SIZE="6">
2	<H2>	<P><B><FONT SIZE="5">
3	<H3>	<P><B><FONT SIZE="4">
4	<H4>	<P><B>,<P><U>, <P><STRONG>, <B><U>, <LI><B>, <DT><B>

4.2.2 요소 식별(Element Identification)

요소 식별은 비주얼 그룹 트리에 문서모델을 적용하여 논리적 구성 요소를 식별한다. 이를 위하여 먼저 문서에 포함된 텍스트를 구분자(delimiter)를 기

1. DOM 트리를 하향식 너비 우선 탐색하면서 비주얼 태그를 검색한다.
2. 검색된 비주얼 태그에 대하여 <GROUP> 요소를 생성하고 해당 텍스트 내용을 "title" 속성 값으로 변환한다.
3. 너비 우선 탐색을 계속하여 현재 비주얼 태그보다 우선순위가 높거나 같은 비주얼 태그 또는 문서의 끝에 도달하면 현재 <GROUP>요소 이후에 위치하는 노드들을 <GROUP> 요소의 자식으로 추가한다.
4. 각각의 <GROUP> 요소에 대해서 1~3번 과정을 반복 적용한다.

그림 6. 제안된 비주얼 그룹화 알고리즘.

```
<p><b><font size="4">Education:</font></b></p>
<ul>
<li><b>Ph.D</b>. in applied mathematics, Cambridge University, Cambridge, UK, 1991</li>
<li><b>M.Sc.</b> in mathematics, University of Waikato, Hamilton, NZ, 1987</li>
<li><b>B.Sc.</b> in mathematics physics, University of Waikato, Hamilton, NZ, 1985</li></ul>
<p><b><font size="4">Employment:</font></b></p>
<ul>
<li><b>Associate Professor</b>,Johns Hopkins University, Baltimore, U.S.A, 2001 to present</li>
```

(a) 복합 비주얼 태그를 포함하는 문서의 예

```
<GROUP TITLE="Education:">
<ul>
<li><b>Ph.D</b>. in applied mathematics, Cambridge University, Cambridge, UK, 1991</li>
<li><b>M.Sc.</b> in mathematics, University of Waikato, Hamilton, NZ, 1987</li>
<li><b>B.Sc.</b> in mathematics physics, University of Waikato, Hamilton, NZ, 1985</li></ul>
</GROUP>
<GROUP TITLE="Employment:">
<ul>
<li><b>Associate Professor</b>,Johns Hopkins University, Baltimore, U.S.A, 2001 to present</li>
```

(b) (a)에 비주얼 그룹화를 적용한 결과

그림 7. 비주얼 그룹화의 예

준으로 토큰화(tokenization)한다. 본 논문에서는 구분자로 ';', ',', 그리고 ':'을 사용한다. 또한, 문서모델에 정의된 키워드 패턴을 포함하는 토큰(token)을 해당 논리적 요소로 식별한다.

제안된 요소 식별 방법은 논리적 구성 요소 사이의 포함 관계 및 계층 구조 정보를 포함하는 문서모델에 기반한다. 따라서 임의의 요소를 식별하는 과정에서 이미 식별된 상위 요소와의 포함관계를 고려하기 때문에 논리적 요소를 정확하게 식별한다. <GROUP> 태그의 속성 TITLE의 값에 해당하는 요소를 식별한 후, <GROUP>가 둘러싸는 영역에 대하여 해당 요소의 자손으로 올 수 있는 요소만을 식별한다.

예를 들어, "university"라는 키워드를 포함하는 토큰이 어느 그룹에도 속해 있지 않다면 요소 ORGANIZATION으로 식별된다. 그러나, 키워드 "university"를 포함하는 토큰이 임의의 그룹에 속해 있으며 해당 그룹에 대응하는 논리적 요소가 요소 ORGANIZATION을 포함할 수 없다면 해당 토큰은 요소로 식별되지 않는다. 이와 같이 제안된 방법은 문서모델에 정의된 논리적 계층구조에 기반하여 단계적으로 요소를 식별한다. 만일 단순히 키워드만으로 요소를 식별한다면 다수의 잘못된 요소를 식별하여 정확한 구조의 XML 문서를 생성할 수 없을 것이다. 제안된 요소 식별 방법에 대한 자세한 기술은 그림 8과 같다.

1. 비주얼 그룹 트리를 너비 우선 탐색하면서 2번 또는 3번 과정을 적용하면서 문서모델에 정의된 요소를 식별한다.
2. 비주얼 그룹화 단계에서 그룹화 된 <GROUP> 태그의 경우에는 다음을 수행한다.
   2.1 <GROUP>의 TITLE 속성 값에 해당하는 논리적 요소를 식별한다. 즉, 식별된 요소 이름에 해당하는 태그를 생성하며 속성 TITLE의 값을 생성된 요소의 속성 VAL의 값으로 변환한다.
   2.2 <GROUP> 태그로 둘러싸인 영역에 대하여 식별된 요소의 자손으로 올 수 있는 요소를 식별한다.
3. 그룹화 되지 않은 영역에 대하여 다음의 요소식별 과정을 적용한다.
   3.1 토큰이 포함하는 키워드를 기반으로 논리적 요소를 식별한다. 이때 해당 토큰을 식별된 논리적 요소를 이름으로 하는 태그의 속성 VAL의 값으로 넣어준다.
   3.2 만일 한 개의 토큰으로부터 두 개 이상의 요소가 식별되면 수식관계를 고려하여 단일의 요소로 식별한다.
   3.3 요소가 식별이 되지 않으면 해당 텍스트 값은 태그 <TEXT>의 속성 VAL의 값으로 변환한다.

그림 8. 요소 식별 알고리즘

그림 9는 요소 식별 과정의 예를 보여준다. 우선 그림 9(a)와 같이 "education"은 상위에 식별된 요소가 없기 때문에 키워드만으로 요소 EDUCATION으로 식별된다. 이때 요소 EDUCATION의 속성 VAL은 토큰의 내용을 값으로 갖는다. 그림 9(b)에서는 "ph.d"가 요소 DEGREE로 식별이 될 수 있다. 상위 요소를 보면 요소 EDUCATION이 식별이 되었고 문서 모델에 이와 같은 포함관계가 명시되어 있기 때문에 요소 DEGREE로 식별된다. 마찬가지로 요소 EDUCATION의 하위 요소로 요소 ORGANIZATION, CITY, NATION 등이 식별된다.

한편, 한 개의 토큰 내에서 두 개 이상의 요소가 식별될 때, 두 요소 중에서 하나의 요소가 다른 하나의 요소를 수식하는 관계가 발생할 수 있다. 예를 들어, 요소 CITY와 ORGANIZATION가 순차적으로 식별되었거나 ORGANIZATION과 CITY 사이에 문자열 "of"가 존재할 경우, CITY가 ORGANIZATION을 수식하는 관계가 된다. 이러한 경우, 두 개의 요소로 식별하지 않고 하나의 요소로 간주한다. "boston university"의 경우, "boston"이 요소 CITY로 식별되며 "university"가 ORGANIZATION으로 식별 가능하지만 이를 통합하여 단일의 요소 ORGANIZATION으로 식별한다. 마찬가지로 "assistant professor"의 경우, "assistant"와 "professor"가 각각 요소 JOBTYPE으로 식별되지만 "assistant"가 "professor"를 수식하는 관계가 성립하므로 하나의 요소로 처리한다.

4.2.3 논리적 그룹화

논리적 그룹화는 전 단계에서 식별된 논리적 요소를 포함하는 비주얼 그룹 트리를 대상으로 보다 정교한 계층구조를 생성한다. 특히 제안된 논리적 그룹화는 여러 요소들이 반복이 될 때 처음에 반복이 되는 요소가 반복되는 내용들을 대표할 수 있다는 점에

```
<GROUP><EDUCATION VAL="Education"/>
   <UL>
      <LI>Ph.D. in applied mathematics, Cambridge University, Cambridge, UK, 1991</LI>
      <LI>M.Sc. in mathematics, University of Waikato, Hamilton, NZ, 1987 </LI>
   </UL>
</GROUP>
```

(a) 첫 번째 레벨의 요소 식별

```
<GROUP><EDUCATION VAL="Education"/>
   <UL>
      <LI>
         <DEGREE VAL ="Ph.D. in applied mathematics"/>
         <ORGANIZATION VAL ="Cambridge University,"/>
         <CITY VAL="Cambridge,"/>
         <NATION VAL="UK,"/>
         <TEXT VAL="1991"/>
      </LI>
      <LI>
         <DEGREE VAL ="M.Sc. in mathematics" />
         <ORGANIZATION VAL="University of Waikato," />
         <CITY VAL="Hamilton,"/>
         <NATION VAL="NZ,"/>
         <TEXT VAL="1987"/>
      </LI>
   </UL>
</GROUP>
```

(b) 두 번째 레벨의 요소 식별

그림 9. 요소 식별의 예

기인한다. 제안된 방법에 대한 자세한 기술은 그림 10과 같다.

<LI>, <DD> 등의 HTML 리스트 아이템 태그에 포함되어있는 내용은 논리적으로 독립된 단위로 간주하여 첫 번째 요소를 부모로 하는 단일의 그룹을 형성한다. 또한, 반복되는 논리적 요소를 기준으로 계층 구조를 생성한다. 논리적 계층 구조를 정확히 생성하기 위하여 반복되는 요소나 리스트 아이템 태그를 기준으로 그룹화할 때 그룹이 제안된 문서 모델에 부합하는지의 여부를 검사한다.

예를 들어, 그림 9(b)와 같이 리스트 아이템 태그인 <LI>가 다수의 요소를 포함할 경우, 그림 11과 같이 첫 번째 요소를 부모로 그룹화할 수 있다. 그림 12(a)의 경우, 요소 EDUCATION은 반복되는 요소인 DEGREE를 자식 요소로 갖는다. 제안된 방법은 요소 DEGREE를 기준으로 {DEGREE, ORGANIZATION, THESIS}와 {DEGREE, ORGANIZATION, GPA}의 두 그룹으로 나눈다. 또한, DEGREE는 기술된 문서모델에서 요소 ORGANIZATION, THESIS, GPA를 모두 자식 요소로 가질 수 있기 때문에 그림 12(b)의 계층 구조를 생성한다. 그러나, 그림 13과 같이 요소 SKILL은 반복되는 요소인 PROGRAMMING을 자식 요소로 갖지만 문서 모델에 요소 PROGRAMMING, OS, 그리고 TOOL은 서로 형제 관계로 정의되어 있기 때문에 더 이상 그룹화되지 않는다.

4.3 후처리

후처리 과정은 논리적 구조분석 과정에서 처리하지 못한 요소를 추가로 식별하며 최종적으로 XML 문서를 생성한다. 문서 모델에 필수 요소로 기술되어 있지만 논리적 구조분석 과정에서 식별하지 못한 부

1. 비주얼 그룹 트리를 하향식 너비 우선 탐색하면서 반복되는 자식요소를 포함하는 노드 또는 HTML 리스트 아이템 태그를 검색하여 2번 또는 3번 과정을 적용한다.
2. HTML 리스트 아이템 태그인 경우 첫 번째 자식 요소를 부모로 하는 그룹을 형성한다.
3. 노드의 자식 중에 반복되는 요소가 존재할 경우, 문서 모델에 부합한다면 반복되는 요소를 부모로 하고, 반복되는 요소 사이에 있는 노드들을 자식으로 하는 계층구조를 생성한다.

그림 10. 논리적 그룹화 알고리즘

```
<GROUP><EDUCATION VAL="Education"/>
    <UL>
        <LI>
            <DEGREE VAL ="Ph.D. in applied mathematics">
                <ORGANIZATION VAL ="Cambridge University,"/>
                <CITY VAL="Cambridge,"/>
                <NATION VAL="UK,"/>
                <TEXT VAL="1991"/>
            </DEGREE>
        </LI>
        <LI>
            <DEGREE VAL ="M.Sc. in mathematics">
                <ORGANIZATION VAL="University of Waikato," />
                <CITY VAL="Hamilton,"/>
                <NATION VAL="NZ,"/>
                <TEXT VAL="1987"/>
            </DEGREE>
        </LI>
    </UL>
</GROUP
```

그림 11. 리스트 아이템 태그에 기반한 논리적 그룹화의 예

```
<EDUCATION val="Education">
    <DEGREE val="Ph.D in applied mathematics"/>
    <ORGANIZATION val="Cambridge University"/>
    <THESIS val="Semistructure" />
    <DEGREE val="Ph.D in applied mathematics"/>
    <ORGANIZATION val="Cambridge University"/>
    <GPA val="3.6/4.0"/>
</EDUCATION>
```

(a) 반복되는 요소를 포함하는 문서의 예

```
<EDUCATION val="Education">
    <DEGREE val="Ph.D in applied mathematics">
     <ORGANIZATION val="Cambridge University"/>
     <THESIS val="Semistructure" />
    </DEGREE>
    <DEGREE val="Ph.D in applied mathematics"/>
     <ORGANIZATION val="Cambridge University"/>
     <GPA val="3.6/4.0" />
    </DEGREE>
</EDUCATION>
```

(b) (a)에 논리적 그룹화 과정을 적용한 결과.

그림 12. 반복되는 요소에 기반한 논리적 그룹화의 예.

구조를 추가하거나 이전 단계에서 식별하지 못한 요소를 식별한다. 예를 들어 PUBLICATION과 같은 요소의 경우, 저자, 타이틀, 페이지번호 등의 정보를 포함한다. 그러나 키워드 기반으로는 이러한 요소를

```
<SKILLS val=" Other Skills">
 <PROGRAMMING val=" Matlab."/>
 <OS val=" Operating Systems Unix"/>
 <OS val=" MS-DOS"/>
 <PROGRAMMING val=" Matlab"/>
 <TOOL val=" LaTeX"/>
 <TOOL val=" Framemaker 5.0"/>
</SKILLS>
```

그림 13. 논리적 그룹화가 적용되지 않는 예

추출해 내기 어렵다. 또한, 요소 EDUCATION는 자식으로 졸업한 연도를 나타내는 요소 DATE를 포함할 수 있는데, 이것은 정규식을 인식해야만 추출해 낼 수 있는 정보이다.

논리적 구조분석 과정에서도 이렇게 자세하게 처리해 줄 수도 있지만, 모든 텍스트를 대상으로 정규표현을 인식하여 추출하도록 하면 시스템에 부하가 많이 걸리고, 모든 부분에서 같은 정규표현식이 적용되는 것도 아니기 때문에 후처리 과정을 두어 처리한다. 우선 정규 표현식에 대해 상위 노드부터 검색하여 정규 표현식이 있을 수 있는 노드만을 검색하여 정규 표현식을 적용할 텍스트를 찾는다. 정규 표현식을 적용할 수 있는 노드가 검색이 되면 각 노드들에 정규 표현식을 적용하여 노드를 식별한다. 이러한 정규 표현식 식별 과정이 끝나면 전술한 논리적 그룹화를 다시 한 번 적용한다.

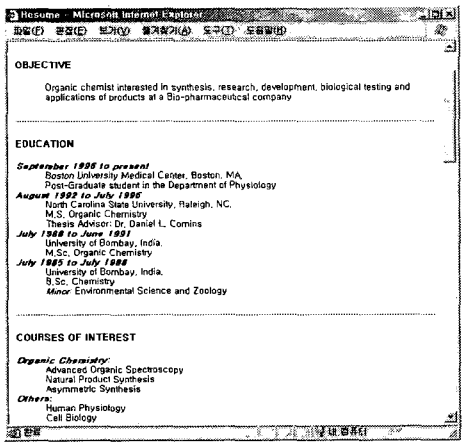
한편, 일반적으로 사용자가 문서를 처음부터 끝까지 읽어가는 순서는 논리 구조 트리를 깊이 우선 탐색하는 과정으로 볼 수 있다. 제안된 방법은 논리 구조 트리를 구성하는 단말 노드와 중간 노드 각각에 대하여 서로 다른 방법을 적용하여 XML 문서를 생성한다. 먼저 논리 구조 트리를 깊이 우선 탐색하면서 중간 노드를 만나면 해당 레이블을 이름으로 갖는 엘리먼트의 시작 태그를 출력하며 해당 노드를 벗어날 때 끝 태그를 출력한다. 한편 단말 노드는 문서의 텍스트 영역과 직접적인 대응 관계를 갖는다. 따라서 단말 노드를 만나면 먼저 해당 레이블의 이름을 갖는 엘리먼트의 시작 태그를 출력하고, 해당 텍스트와 끝 태그를 출력함으로써 엘리먼트를 생성한다.

5. 실험 결과

제안된 방법의 성능을 평가하기 위하여 Chung 등의 연구에서 사용한 50개의 이력서 HTML 문서를 대상으로 실험하였다. 실험 결과, 제안된 방법은 그림 14와 같이 HTML 문서로부터 논리적 구조 정보를 추출하여 XML 문서를 생성하였다.

5.1 성능 평가

본 논문에서는 구조분석의 정확성 측면에서 제안된 방법의 성능을 분석하였다. 전술한 바와 같이 제안된 논리적 구조분석 과정은 크게 비주얼 그룹화, 요소 식별, 그리고 논리적 그룹화의 세 단계로 구성



(a) 원본 HTML 문서

(b) 추출된 XML 문서

그림 14. 실험 결과

된다. 따라서, 본 논문에서는 제안된 방법의 성능을 평가하기 위하여 표 4와 같이 비주얼 그룹화의 정확성, 논리적 구성 요소의 식별률, 그리고 논리적 계층 구조의 정확성의 세 가지 평가 기준을 정의한다.

제안된 방법의 성능을 정량적으로 평가한 결과는 표 5와 같다. 먼저 비주얼 그룹화의 정확성 면에서, 제안된 방법은 94.10%의 정확률을 보였다. 비주얼 그룹화의 오류분석은 다음과 같다. 그림 15(a)의 경우 시각적으로 "education", "relevant coursework", 그리고 "technical programming skills"의 세 그룹으로 나눌 수 있다. 반면, 그림 15(b)의 HTML 문서에서는 비주얼 태그 <P><B>를 기준으로 해서 "education", "relevant", "coursework", "technical", "programming", 그리고 "skills"의 6개 그룹으로 나누어진다. 이와 같이 시각적으로 보기에는 그룹화 할 수 있지만, 제안된 비주얼 태그를 적용하여 비주얼 그룹화할 수 없는 경우가 존재하였다.

또한, HTML 문서를 작성자가 임의의 불규칙한 태그를 사용하여 비주얼 그룹을 표현한 경우, 적절히 그룹화되지 않았다. 그림 15(c)와 같이 시각적으로 같은 레벨로 보이는 "education"과 "activities"가 그림 15(d)에서와 같이 서로 다른 종류의 태그인 <DT><B>와 <B>를 사용하였기 때문에 요소 EDUCATION과 요소 ACTIVITIES는 서로 다른 레벨의 그룹으로 그룹화 되었다. 비주얼 그룹화 오류의 대부분은 위 두 가지 경우에 해당하였으며 그 밖에 비주얼 태그가 그룹화 이외의 용도로 사용되어 적절히 그룹화되지 못한 경우가 존재하였다.

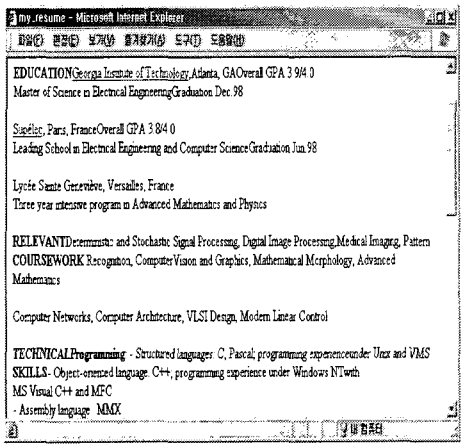
요소의 식별률 면에 있어서 제안된 방법은 표 5와 같이 93.21%의 정확률과 91.62%의 재현율을 보였다. 실험 결과, 식별된 요소의 오류분석은 다음과 같다. 예를 들어, 그림 15의 경우, 비주얼 그룹화가 성공적으로 이루어지지 못하여 제안된 요소 식별 정보를 적용할 수 없었다. 예를 들어, 그림 15(a)는 사람이

표 5. 성능 평가

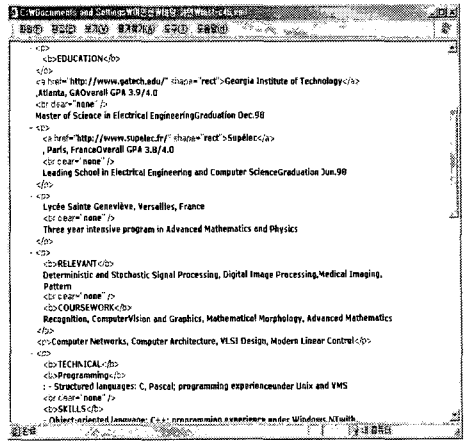
번호	비주얼 그룹화(%)	요소 식별 정확율(%)	요소 식별 재현율(%)	계층 구조 정확성(%)
1	100	93.42	98.61	96.50
2	100	87.27	85.71	85.58
3	100	97.30	100.00	98.61
4	100	100.00	97.30	98.63
5	83.33	93.18	95.35	94.05
6	90	97.14	94.44	95.71
7	84.81	92.73	96.23	94.23
8	100	97.50	97.50	97.47
9	100	98.00	96.08	97.00
10	100	98.04	96.15	97.06
11	100	91.53	96.43	92.73
12	100	89.47	88.54	88.40
13	100	86.11	88.57	86.36
14	100	92.19	86.76	88.98
15	84.61	95.24	97.56	96.30
16	100	100.00	100.00	100.00
17	100	88.37	88.37	87.65
18	100	100.00	91.78	95.00
19	100	81.25	83.87	80.70
20	100	100.00	76.47	86.67
21	100	96.55	90.32	93.22
22	100	87.76	97.73	91.95
23	100	95.83	85.19	92.00
24	100	97.37	92.50	94.81
25	44.44	88.57	93.94	90.63
26	0	79.41	64.29	60.87
27	100	100.00	100.00	100.00
28	0	93.75	100.00	90.00
29	100	96.77	88.24	92.19
30	100	92.75	96.97	94.62
31	100	94.74	94.74	94.59
32	100	96.47	95.35	94.05
33	100	92.00	92.00	91.67
34	100	100.00	100.00	100.00
35	100	93.10	96.43	94.55
36	60	95.12	97.50	96.20
37	75	94.44	94.44	94.29
38	100	80.65	89.29	88.68
39	50	100.00	93.33	96.55
40	100	90.00	79.41	83.61
41	100	93.55	89.23	91.87
42	100	96.00	89.74	91.33
43	0	91.30	89.36	92.13
44	100	88.00	89.36	87.91
45	100	93.62	95.65	94.44
46	100	88.89	90.91	89.29
47	100	88.68	95.92	91.67
48	100	90.48	86.36	87.80
49	100	95.71	90.54	95.04
50	100	94.83	90.60	92.07
평균	94.10	93.21	91.62	92.02

표 4. 성능 평가 기준

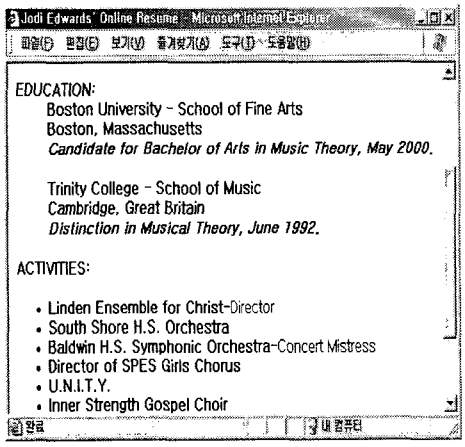
기 준		정 의
비주얼 그룹화의 정확성		정확하게 추출된 비주얼 그룹의 수/추출된 비주얼 그룹의 수
요소의 식별률	정확률	정확하게 식별된 요소의 수/식별된 요소의 수
	재현율	정확하게 식별된 요소의 수/검증용 데이터(groundtruth data)의 요소 수
계층 구조의 정확성		$1-\frac{\text{편집스크립트}}{\text{추출된 계층 구조 + 정확한 계층 구조}} \times 100$



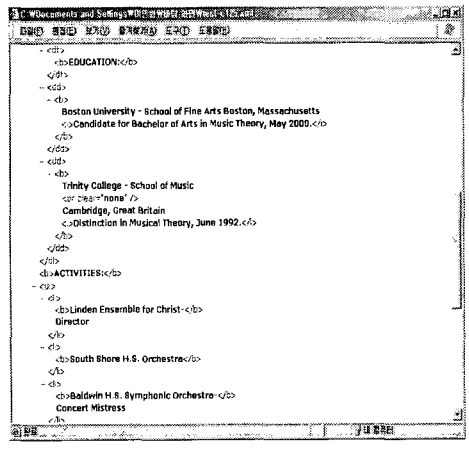
(a) 그룹화 힌트 적용이 어려운 경우



(b) (a)의 HTML 소스 코드.



(c) 불규칙한 그룹화 태그를 사용한 예



(d) (c)의 HTML 소스 코드.

그림 15. 잘못된 비주얼 그룹화의 예.

보기에는 그룹화가 가능하지만 실제 HTML 코드(그림 15(b) 참조)는 적절한 그룹화 태그를 포함하지 않는다. 따라서, 제안된 방법은 그림 16(b)의 검증용 데이터와 다른 그림 16(a)의 결과를 생성하였다. 즉, 그림 16(a)에서 "ms"를 요소 DEGREE로 잘못 식별하였다. 그림 16(b)와 같이 그룹화가 이루어져 SKILL을 부모 요소로 갖는다면 "ms"를 잘못된 요소로 식별하지 않았을 것이다. 또한, 실험에 사용된 데이터 중에는 그림 17과 같이 띄어쓰기 등의 편집 오류에 의하여 요소를 식별할 수 없는 경우가 존재하였다.

요소 식별률 면에 있어서 제안된 방법은 문서 모델에 정의된 요소간의 계층적 구조 정보를 기반으로 요소를 식별하기 때문에 문맥에 따라 다른 의미를 갖는 키워드를 보다 정확히 식별한다. 반면에 문서 모델에 정의하지 않은 계층 구조나 해당 요소의 식별에 필요한 키워드 또는 패턴 정보를 포함하지 않은 경우 요소를 식별하지 못하였다.

마지막으로, 논리적 계층 구조의 정확성을 실험하기 위하여 구조분석의 결과로 생성된 계층 구조와 정확한 구조 사이의 구조적 차이를 계산하였다. 이를 위하여 트리 간의 편집 스크립트(edit script)를 계산하는 기존 연구인 Zhang과 Shasha의 방법[14]을 적용하였다. 한편, [14]는 두 트리 사이의 차이를 나타내기 위하여 노드의 삽입(insert), 삭제(delete), 그리고 갱신(update)의 세 가지 편집 연산(edit operation)을 정의한다. 본 논문은 각 연산의 비용을 1로 가정하였다. 특히, 보다 의미 있는 차이를 계산하기 위하여 추출된 차이를 생성된 트리 구조와 정확한 트리 구조

의 합으로 정규화 하였다.

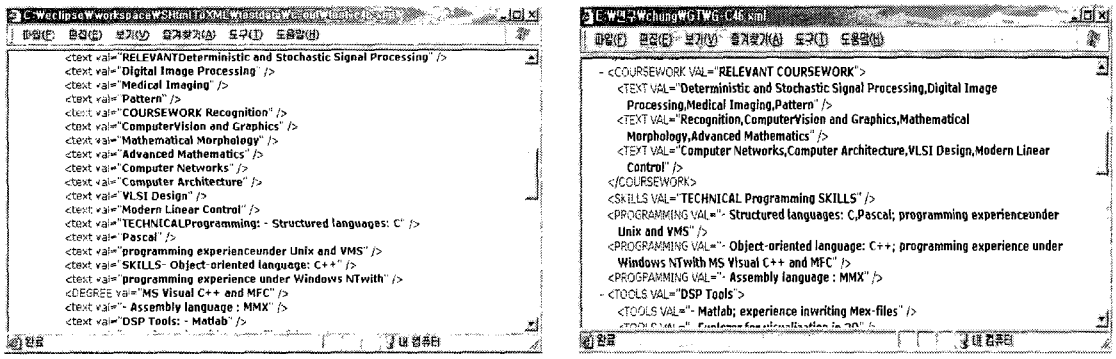
실험 결과, 표 5와 같이 계층 구조의 정확성은 평균적으로 92.02%로 나타났다. 오류의 대부분은 제안된 방법의 비주얼 그룹화와 요소 식별 과정에서 찾을 수 있었다. 우선, 그림 15와 같이 비주얼 그룹화가 적절히 이루어지지 않으면 그림 16에서 설명하였듯이 잘못된 구조를 생성하였다. 마찬가지로 그림 17과 같이 편집 오류로 인해 요소를 식별하지 못하면 해당 요소에 대한 삽입 연산을 적용해야 하기 때문에 구조적인 차이가 발생하였다.

5.2 기존 연구와의 비교

한편 Chung 등은 본 논문과 달리 추출된 XML 문서 트리에서 잘못된 부모-자식 그리고 형제 관계의 수만을 계산함으로서 구조적 정확성을 계산하였다. Chung 등이 제안한 평가기준에 따라 계산할 경우, 제안된 방법은 93.2%의 정확성을 보인다. 한편 Chung 등의 방법은 본 논문과 동일한 실험 데이터를 대상으로 90.2%의 정확성을 보였다.

제안된 방법은 동일한 실험 데이터를 이용한 Chung 등의 방법보다 구조분석의 정확성 측면에서 우수한 결과를 보였다. 이는 제안된 방법이 요소 식별 과정에서 비주얼 그룹화에서 그룹화된 정보를 이용하여 요소들 사이의 부모 자식관계를 고려하기 때문이다. Chung 등의 방법에서도 부모 자식 관계를 고려하기는 하지만 요소를 식별 후에 그룹화 과정에서 자식으로 올 수 있는지 만을 체크하기 때문에 근본적으로 요소 식별 과정에 영향을 미치지는 않는다.

반면, 제안된 방법은 문서 모델에 정의된 부모 자식 관계를 기반으로 잘못된 요소의 식별을 근본적으로 방지하기 때문에 보다 정확한 요소 식별을 지원한다. 예를 들어, 키워드 "ms"는 "master of science" 또는 "microsoft"의 약어로 사용된다. 이때 임의의 한 개의 요소로 식별하는 Chung등의 방법과 달리,



(a) 잘못된 요소 식별 결과 (b) 검증용 데이터(ground-truth data)

그림 16. 비주얼 그룹화 오류에 따른 잘못된 요소 식별의 예

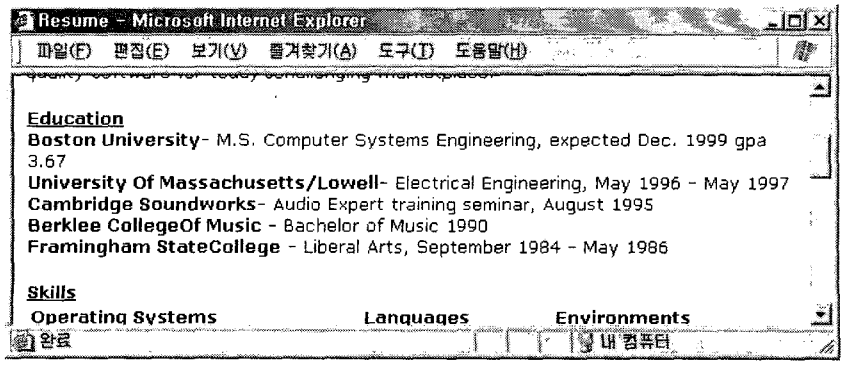


그림 17. 편집 오류로 인하여 요소를 식별할 수 없는 예.

제안된 방법은 문서 모델에 정의된 부모 자식관계를 고려하여 해당 키워드를 적절한 요소로 식별한다. 이미 식별된 부모 요소의 이름이 EDUCATION이라면 해당 키워드를 요소 DEGREE로, 만일 부모 요소의 이름이 EXPERIENCE라면 요소 COMPANY로 식별한다.

한편, Lim과 Ng은 본 논문과 달리 XML 문서를 생성함에 있어 요소 이름으로 해당 텍스트를 그대로 사용한다. 이에 논리적 구조 정보를 포함하는 XML 문서를 생성하는데 제한적이다. 한편 김승원 등은 사용자와의 상호작용을 통해서 요소 이름을 설정하기 때문에 XML 문서를 자동으로 생성하지 못한다.

6. 결론 및 향후 연구 방향

XML은 논리적인 구조 정보를 표현할 수 있으며 이 기종간의 호환이 가능하다는 장점 때문에 전자문서의 표준 포맷으로 널리 사용되고 있다. 따라서 본 논문에서는 웹 문서로부터 XML 문서를 생성하기 위한 논리적 구조분석 방법을 제안한다.

제안된 방법은 정교한 수준의 구조분석을 위하여 문서 모델을 효과적으로 기술할 수 있는 언어인 MEDL을 제안한다. MEDL은 논리적인 계층 구조를 기술하기 위하여 특정 주제에 속하는 문서 집합이 포함할 수 있는 논리적 구성 요소의 종류, 포함관계, 그리고 빈도수 등에 대한 다양한 정보를 기술한다.

제안된 방법은 논리적 계층구조의 분석을 위해 비주얼 그룹화, 요소식별, 그리고 논리적 그룹화의 세 단계로 구성된다. 우선 비주얼 그룹화에서는 시각적으로 그룹화할 수 있는 정보를 갖는 태그들을 비주얼 태그라 정의하고 전처리를 거친 DOM 트리를 하향식 너비 우선 탐색을 적용하여 비주얼 태그들 사이의 내용을 그룹화하여 비주얼 트리를 생성한다. 요소 식별은 비주얼 트리에 문서 모델에 정의된 요소 식별 정보를 적용하여 키워드를 기반으로 하여 요소를 식별한다. 또한, 식별된 요소들의 반복과 리스트 아이템 태그들의 특징을 이용하여 논리적 계층구조로 그룹화한다.

제안된 방법의 성능을 분석하기 위해 비주얼 그룹화의 정확성, 논리적 구성 요소의 식별률, 그리고 논리적 계층 구조의 정확성의 세 가지 평가 기준으로 성능을 평가한 결과, 기존 연구와 비교하여 효율적인 논리적인 구조분석을 수행하였다. 또한 제안된 방법은 MEDL에 기반한 문서 모델에 따라 논리적 구조분석을 수행하기 때문에 생성된 XML 문서는 문서 모델에 포함된 요소로 구성된 유사한 논리적 구조를 갖는다. 따라서 추출된 XML 문서는 재사용성 측면에서도 효과적이다.

한편, 제안된 방법의 오류를 분석한 결과, 비주얼 그룹화가 성공적으로 이루어지지 않을 경우, 잘못된 계층 구조를 생성함을 볼 수 있었다. 또한, 정확한 요소 식별을 위해서 보다 정교한 수준의 논리적 요소 기술 방법이 요구된다. 따라서 향후 본 연구에서는 보다 정교한 수준의 비주얼 그룹화 및 요소 식별 방법을 연구할 계획이다.

참 고 문 헌

[1] World Wide Web Consortium, (Hypertext Markup Language (HTML) 4.0, W3C Recommendation, http://www.w3c.org/TR/REC-html40, 1999.

[2] World Wide Web Consortium, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, http://www.w3c.org/TR/REC-xml, 2000.

[3] K. M. Summers, "Toward a Taxonomy of Logical Document Structures," Proc. Dartmouth Inst. for Advanced Graduate Studies (DAGS '95) pp. 124-133, May 1995.

[4] Seung Jin Lim and Yiu-Kai Ng, "A Heuristic Approach for Converting HTML Documents to XML Documents," Computational Logic, pp. 1181-1196, 2000.

[5] David Taniar, Y. Jiang, J. Wenny Rahayu, and L. Bishay, "Structured Web Pages Management for Efficient Data Retrieval," Proc. Int'l Conf. Web Information Systems Engineering, pp. 97-104 2000.

[6] Christina Yip Chung, Michael Gertz, and Neel Sundaresan, "Quixote: Building XML Repositories from Topic Specific Web Documents," Proc. Int'l Conf. WebDB, pp. 103-108, 2001.

[7] Wei Han, David Buttler, and Calton Pu,

"Wrapping Web Data into XML," SIGMOD Record, vol. 30, no. 3, pp. 33-38, 2001.

[8] 김승원, 민준기, 정진완, "사용자와의 상호작용을 통한 HTML 문서의 XML 문서로의 변환", 정보과학회추계학술발표대회 논문집, pp. 103-105, 2002.

[9] 오금용, 황인준, "유사 패턴을 갖는 HTML 문서의 XML 자동 변환", 한국정보처리학회논문지, 제9-D권, 제3호, pp. 355-364, 2002.

[10] Xiaofeng Meng, Haiyan Wang, Dongdong Hu and Chen Li, "A Supervised Visual Wrapper Generator for Web-Data Extraction," Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03), pp. 657-662, 2003.

[11] Arvind Arasu and Hector Gracia-Molina, "Extracting Structured Data from Web Pages," Proc. SIGMOD Int'l Conf. on Management of data, pp. 337-347, 2003.

[12] World Wide Web Consortium, XML DOM Level 3 Core, http://www.w3.org/TR/2003/WD-DOM-Level-3-Core, W3C Candidate Recommendation, Feb. 2003.

[13] Dave Raggett, "Clean up Your Web Pages with HP's HTML Tidy," Computer Networks and ISDN Systems, vol. 30, pp. 730-732, Apr. 1998.

[14] K. Zhang and D. Shasha, "Simple Fast Algorithms for the Editing Distance between Trees and Related Problems," SIAM Journal on Computing, vol. 18, no. 6, pp. 1245-1262, 1989.

이 민 형

2003년 연세대학교 컴퓨터과학과 졸업(학사)
2003년~현재 연세대학교 컴퓨터과학과 석사과정

관심분야 : 웹문서 분석, 정보 추출 및 통합, XML

이 경 호

1995년 연세대학교 전산과학과 졸업(학사)
1997년 연세대학교 컴퓨터과학과 졸업(석사)
2001년 연세대학교 컴퓨터과학과 졸업(박사)
2001년 National Institute of Standards and Technology(NIST) 객원연구원
2002년~현재 연세대학교 컴퓨터산업공학부 조교수
관심분야 : 멀티미디어 문서처리, XML, 웹 서비스