유사어 벡터 확장을 통한 XML태그의 유사성 검사

(Similarity checking between XML tags through expanding synonym vector)

이 정 원 † 이 혜 수 †† 이 기 호 †††

(Jung-Won Lee) (Hye-Soo Lee) (Kiho Lee)

요 약 XML(eXtensible Markup Language)문서가 웹 문서의 표준으로 자리 매김 할 수 있는 가장 큰 성공요인은 사용자가 문서 타입을 기술할 수 있는 유연성(flexibility)이다. 그러나 XML의 유연성으로 야기되는 문제점은 동일한 의미를 표현하기 위해 XML문서 작성자마다 서로 다른 태그명과 구조를 사용한다는 점이다. 즉 서로 다른 태그 집합, 요소(element), 속성(attribute)에 대한 서로 다른 이름 또는 다른 문서 구조로 인해 다른 태그로 표현된 문서는 서로 다른 부류의 문서로 간주되기 쉽다. 따라서 본 논문은 XML태그에 내재된 의미 정보(semantic information)와 구조 정보(structured information)를 추출하여 의미적으로 최대한 유사한 동의어로 확장하고, XML문서의 확장된 태그간의 의미적 유사도를 비교 분석할 수 있는 개념 기반의 태그 패턴 매처(Tag Pattern Matcher)를 설계 구현하였다. 두 XML문서의 태그간의 의미적 유사도에 가중치를 부여하여 기존의 비구조적인(semi-structured) 문서를 위한 벡터 스페이스 모델(vector space model)을 확장함으로써 두 XML문서가 유사한지를 파악할 수 있다.

키워드 : XML, 정보 검색, 문서 처리, 문서 분석

Abstract The success of XML(eXtensible Markup Language) is primarily based on its flexibility : everybody can define the structure of XML documents that represent information in the form he or she desires. XML is so flexible that XML documents cannot be automatically provided with an underlying semantics. Different tag sets, different names for elements or attributes, or different document structures in general mislead the task of classifying and clustering XML documents precisely. In this paper, we design and implement a system that allows checking the semantic-based similarity between XML tags. First, this system extracts the underlying semantics of tags and then expands the synonym set of tags using an WordNet thesaurus and user-defined word library which supports the abbreviation forms and compound words for XML tags. Seconds, considering the relative importance of XML tags in the XML documents, we extend a conventional vector space model which is the most generally used for document model in Information Retrieval field. Using this method, we have been able to check the similarity between XML tags which are represented different tags.

Key words : XML, Information Retrieval, Document Processing, Document Analysis

1. 서 론

월드 와이드 웹(World Wide Web)은 어떠한 주제와 관련된 광범위한 범위의 정보를 보급할 뿐 아니라, 상호 교류를 위한 중요한 매체로써 급속히 발전하였다. 사람들의 예상대로 앞으로 십여 년 후에는 인간이 필요한 대부분의 정보는 웹에서 얻을 수 있을 것이다. 현재 웹에는 대략 3억 개의 웹 페이지가 존재하며, 매일 100만 개의 웹 페이지가 추가되고 있다[6]. 이와 같은 방대한 웹을 대상으로 더욱 효율적이면서 효과적인 방법으로 검색하기 위해서 웹 크롤러(web crawler), 검색 엔진, 야후와 같은 웹 디렉토리는 정보 검색(*information retrieval*)을 위한 최상의 기술로 구성되어 있다. 그러나 데이타양이 급속히 증가함에 따라 여러 가지 문제를 야기하고

† 학생회원 : 이화여자대학교 컴퓨터학과
jungwony@ewha.ac.kr
†† 비 회 원 : 이화여자대학교 컴퓨터학과
hyesoo@samsung.com
††† 종신회원 : 이화여자대학교 컴퓨터학과 교수
khlee@ewha.ac.kr
논문접수 : 2001년 1월 26일
심사완료 : 2002년 6월 19일

있다.

XML(eXtensible Markup Language)의 출현은 이러한 문제에 대한 부분적인 해결책을 제시한다[1]. 기존의 HTML태그는 주로 문서를 웹브라우저 상에 보여주는 형식을 기술하는데 사용하는 반면에, XML 태그는 데이타 그 자체에 대한 설명으로써 태그마다 의미를 지니고 있다. 그러나 XML의 특징인 자기 서술적인 기능(self-describing)[2]으로 인해 야기되는 문제점은 동일한 의미를 표현하기 위해 XML문서 작성자마다 서로 다른 태그명과 구조를 사용한다는 점이다[3]. 예를 들면 XML문서 작성자에 따라서 '책의 저자' 라는 의미를 표현할 때, <author>, <writer>태그 등을 사용할 수 있다. '제품' 이라는 의미를 표현할 때, <products>, <goods>, <commodity>등의 단어를 사용해서 다르게 표현한다. 이와 같이 <author>, <writer>등의 표현은 인간이 보면 바로 동일한 의미로 인식할 수 있지만 애플리케이션에서는 동일한 의미로 파악하는 것이 불가능하다. 따라서 같은 의미를 다른 태그로 표현된 문서는 서로 다른 부류의 문서로 간주되기 쉽다. 즉 서로 다른 태그 집합, 요소(element) 또는 속성(attribute)에 대한 서로 다른 이름 혹은 서로 다른 문서 구조 때문에 정보에 접근하는데 있어서 방해가 된다. 예를 들면 키워드 기반 매칭 기법을 사용하여 검색하였을 경우에 정확하게 일치하지 않을 경우에는 매칭되는 결과가 없다든지 아니면 관련이 없는 너무나 많은 문서를 검색 결과로 던져 주는 경우가 있다. 그러나 태그의 의미를 고려하여 어느 정도 유사한지를 판단할 수 있다면 이러한 검색의 질을 높일 수 있다는 사실은 자명한 일이다. 따라서 본 논문은 XML태그에 내재된 의미정보(semantic information)를 추출하여 시소러스인 WordNet과 사용자 정의 유사 용어 사전을 기반으로 각 태그를 최대한 의미적으로 유사한 동의어로 확장하여 두 XML문서의 확장된 태그간의 의미적 유사도를 비교 분석할 수 있는 태그 패턴 매처(Tag Pattern Matcher)를 설계 구현하였다.

본 논문은 2장에서는 정보검색 모델, 시소러스를 살펴보고, 3장에서는 XML태그간의 유사도를 측정하기 위한 시스템을 설계, 구현하고 예제를 통해 검증한다. 마지막으로 4장에서는 결론을 맺고 향후 연구 방향을 제안한다.

2. 관련 연구

HTML과 XML과 같은 반구조적(semi-structured)인 텍스트 문서의 증가로 인해 효과적이고 효율적인 정보 검색과 필터링을 지원해야하는 필요성이 더욱 증가되고 있다. 정보 검색분야에서 텍스트 문서의 군집화(clustering) 또는 분류(classification)에 대한 연구는 광범위하게 진행되어 왔다[4]. 그러나 이와 같은 기존의 방법에서는 구조화된 문서에 대해서는 잘 수행되지 않는다. 그 이유는 기존의 방법들은 비구조화된(non-structured) 데이타를 위해 고안된 모델이기 때문이다. 반구조적인(semi-structured) 문서인 XML문서에서 태그의 구조 정보와 의미 정보가 차지하는 중요도를 전혀 고려하지 않았다. 기존의 정보 검색모델을 살펴보면, 크게 15가지 모델로 분류할 수 있다[5]. 고전적인 모델인 불린(boolean), 벡터(vector), 확률(probabilistic)모델, 이러한 세 가지 타입이외에 각 타입에 대한 대안적인 모델 파라다임이 수십 년 동안 제안되었다. 불린 모델의 대안으로는 퍼지(fuzzy), 확장된 불린 (expanded boolean), 벡터 모델의 대안으로는 일반화된 벡터(generalized vector), latent semantic indexing, 신경망(neural network)모델, 확률 모델의 대안으로는 추론망(inference network), 믿음망(belief network)모델이 있다. 이러한 15가지 모델의 정보검색 모델의 분류를 도식화하면 다음과 같다[5].

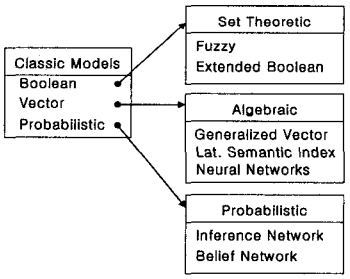


그림 1 정보 검색 모델의 분류

각 타입의 대안적인 모델은 비교적 최근에 제안된 모델로써 아직 실제 시스템에 적용되지 않은 모델도 있으며, 성능이 확실하게 검증되고 있지 않은 모델도 있기 때문에 성능 비교가 불가능하다. 고전적인 모델의 성능 비교는, 일반적으로 불린 모델이 가장 취약한 모델로써 주된 요인은 부분 매칭이 불가능하다는 점이다. 관련성이 있거나 아니면 없는 문서로 판단되기 때문에 낮은 성능을 가져올 수 밖에 없다. 확률모델과 벡터 모델간의 성능은 약간의 논쟁의 여지가 있다. Crof가 몇 가지 실험을 통해서 확률모델이 더욱 나은 성능을 보인다고 제안하였다. 그러나 그 후에, Salton과 Buckley가 다른 측정방법을 통하여 벡터 모델의 성능이 확률모델보다 훨씬 뛰어난 검색 결과를 가져온다고 검증하였다. 이 결과가 여러 연구자들로부터

인정을 받아 벡터 모델이 대중적으로 가장 널리 사용되고 있다[5]. 벡터 공간 모델에서는 문서를 벡터 공간상에서 한 점으로 취급한다. 벡터 공간은 문서 집합에 나타나는 색인어에 의해 결정되고, 각 문서는 색인어가 차지하는 정도를 가중치로 부여하여 각 문서의 용어벡터로 표현한 것이다. 벡터 스페이스 모델로 XML문서를 표현할 때의 문제점은 문서 내에서 태그가 지니는 의미 정보 및 구조 정보를 전혀 반영하고 있지 않다는 점이다. 단순히 색인어가 그 문서에서 차지하는 빈도수만을 고려하기 때문에 동일한 내용을 서로 다른 태그로 표현하였다면 이는 벡터 스페이스 모델에서 아무런 의미를 갖지 못한다. 따라서 본 논문에서는 자연언어 처리 분야에서 사용되는 시소러스를 이용하여 각 태그의 의미간의 유사성을 반영하였다.

시소러스는 단어를 의미에 따라 분류 배열하고 각 단어에 대해 동의어, 유의어, 상위어, 하위어, 반의어, 대의어 등을 기술한 사전이다. 정보 검색분야에서는 용어 차이에 의한 검색 실패를 동의어 관계 정보로 방지한다는 목적에서 일찍부터 시소러스가 이용되어 왔다. 최근에는 자연언어처리에서 단어와 단어의 유사도를 계산하는 데 이러한 시소러스가 중요한 역할을 하고 있다. 프린스턴(Princeton)대학의 Miller가 만든 WordNet은 단어형이 아닌 단어의 의미를 구성요소로 하였다는 특징을 가진 시소러스로서 다의성과 동의관계를 이용하여 의미를 최대한 정확히 표현하고 있다. 또한 의미망형식으로 단어들을 색인하여 저장하고 있기 때문에 일반 사전의 단편적인 알파벳식 사전보다 단어들간의 연관성을 알아내는데 훨씬 유용하다. 본 논문에서는 이러한 특성을 가진 WordNet의 동의어 집합을 이용하여 XML문서의 태그를 확장하려고 한다.

3. XML태그의 의미 기반 유사도 검사를 위한 시스템 설계

본 장에서는 XML태그간의 의미적 유사도를 파악하기 위해서 XML태그의 의미 기반 유사도 검사 시스템을 설계한다. 시스템은 크게 '문서로부터 정보 추출' 부분, '동의어 벡터 생성' 부분, '유사도 측정' 부분과 '개념 지식' 부분의 네 부분으로 구성된다. 먼저 전체 시스템의 구성과 흐름을 설명한 후, 각 부분별로 그 설계 내용과 동작을 설명한다.

3.1 전체 흐름도

크게 정보 추출, 동의어 벡터 생성, 유사도 측정 그리고 개념 지식으로 4부분으로 구성된다. 전체적인 구성도는 [그림 2]와 같다.

정보 추출기는 입력된 두 XML문서에서 내용(content)

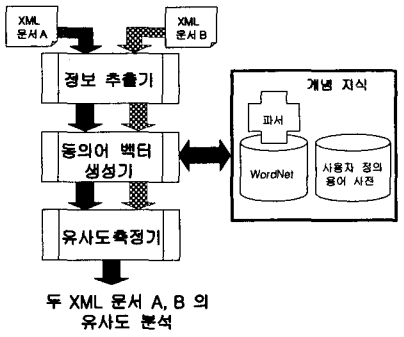


그림 2 전체적인 시스템 구조도

과 태그를 분리하여 태그부분만을 추출하고, 추출된 태그에서 불용어 제거, 스테밍 등의 작업을 통해 스템된 태그 트리와 컨텐츠-태그 테이블을 생성한다. 정보 추출기는 전체 시스템에서 전처리 과정에 해당된다.

동의어 벡터 생성기는 정보 추출기 결과인 스템된 태그 트리를 가지고 WordNet시소러스와 사용자 정의 유사 용어 사전을 이용하여 각 태그의 동의어 벡터(synonym vector)를 생성한다.

유사도 측정기는 두 XML문서의 태그에 있어서 의미적으로 유사한 태그를 파악하고, 두 XML문서의 태그의 유사정도를 측정한다.

3.2 개념 지식 모듈

정보 검색분야에서는 용어 차이에 의한 검색 실패를 동의어 관계 정보로 방지한다는 목적에서 일찍부터 시소러스가 이용되어 왔다. 한편 최근의 자연언어처리에서는 단어와 단어의 유사도를 계산하는 데 이러한 시소러스가 중요한 역할을 하고 있다[6].

현재 가장 널리 사용되는 시소러스로는 1990년 구축한 프린스턴(Princeton) 대학의 Miller가 만든 WordNet이 있다[7,8,9]. WordNet의 주요 특징은 단어형이 아닌 단어의 의미를 구성요소로 동의어 집합(synonym set, synset)이 기본 단위이며, 단순히 단어보다는 개념(concept)으로 인덱스되어 있다. 이러한 특징으로 인해 본 논문에서 WordNet을 개념지식으로 사용하였다. 본 논문에서 사용된 WordNet 버전은 WordNet1.6 으로써 121,962개의 단어와 99,642개의 개념으로 구성되어 있다[7]. 이와 같은 WordNet의 여러 장점에도 불구하고 XML문서의 태그를 확장하기 위해 WordNet만을 사용하기에는 다음과 같은 문제가 발생한다.

첫째, WordNet은 넓은 범위에 해당되는 단어를 포함하고 있지만 특정 분야의 전문적인 단어에 있어서는 부족하다[9].

둘째, WordNet에서는 사용자가 정의하여 사용한 생략어, 합성어, 두문자어 등을 지원하지 못한다. XML문서의 태그를 살펴보면, 단어의 길이가 길 때, 일반적으로 그 단어를 생략해서 표현하는 경우가 많다. 예를 들면, <section> 대신에 <sect>으로 표현한다든지, 또는 <bibliographies>대신에 <biblio>라고 표현한다. 이러한 경우는 WordNet에서 그 동의어를 찾을 수가 없다. 책의 발행일이란 의미로 <publication date>라는 단어를 <pubdate>라고 표현한다든지, 문단을 의미하는 <paragraph>를 <para>라고 표현하는 등의 합성어 등도 지원하지 못한다.

셋째, WordNet에서는 두문자어를 지원하지 못한다. 예를 들면 책의 시리얼 숫자를 의미하는 <ISBN>, 페이지를 나타내는 <pp> 등의 단어는 자연 언어를 대상으로 하는 WordNet에서는 지원하지 않는다.

위와 같은 문제점을 해결하기 위해서 이들 단어에 대한 동의어 집합을 지원할 수 있는 온톨로지가 필요하다. 본 논문에서는 이를 사용자 정의 유사 용어 사전(user-defined synonym word library)이라 한다. 사용자 정의 유사 용어 사전의 필요성은 자연언어를 대상으로 하는 WordNet에서 지원하지 못하는 단어를 대상으로 의미상 동의어를 이끌어 내기 위함이다. 모든 도메인을 대상으로 사용자 정의 유사 용어 사전을 구축할 수 없기 때문에 먼저 책 관련 도메인을 선정하였다. 책 관련 도메인을 선정한 이유는 다음과 같다.

XML의 근원인 SGML(Standard Generalized Markup Language)의 주요 목적은 웹 출판을 위한 마크업 언어로써 SGML 산업-표준 DTDs를 개발하기 위한 많은 노력이 있었다. 따라서 책, 기사, 메뉴얼 등을 웹에 출판을 위한 목적으로 사용할 때는 SGML 산업-표준 DTDs를 따른다. 또한 산업-표준 DTDs가 XML 버전으로 진행중이다[10]. 따라서 이러한 DTDs에서 사용하는 태그를 기준으로 사용자 정의 유사 용어 사전을 구축한다면 더욱 정확하고 객관적인 자료가 될 것이다. 따라서 본 논문에서는 산업-표준 DTDs 모델인 ISO 12083, DocBook[11], TEI-Lite, MIL-STD-38784, HTML을 기준으로 동일한 기능을 수행하는 요소와 속성을 모아서 동의어 집합을 구성하였다. 다음의 예는 동일한 기능을 수행하는 태그로써 DTD모델에 따라 다르게 표현된 예이다.

[표1]에서 문단을 의미하는 'paragraph'는 DTD모델에 따라 <para>, <p>로 표현된다. 따라서 <paragraph>의 의미로 <para>, <p>태그를 사용했을 경우에 이를 동일한 의미의 태그로 파악하기 위해서, 위에서 설명한 5가지 DTD모델에서 동일한 의미를 타나내는 태그를

표 1 DTD 모델별 동일한 의미의 태그

ISO12083	DocBook	TEI-Lite	MIL-STD-38784	HTML
P	P	P	para	P
emph	emphasis	emph	emphasis	EM
item	listItem	Item	Item	LT
cell	entry	cell	entry	TD

파악하여 사용자 정의 유사 용어 사전을 구축하였다. 이렇게 구축된 사용자 정의 유사 용어 사전을 이용한다면 <p>태그가 들어왔을 때, <p>태그의 동의어로써 <para>, <paragraph>태그로 확장 가능하다.

3.3 정보 추출기

정보 추출기는 태그의 패턴을 비교하기 위한 전처리 작업으로, 입력은 유사도를 판단하고자 하는 두 XML문서이고, 결과는 태그와 컨텐츠의 쌍으로 이루어진 컨텐츠-태그 테이블과 스템화된 태그 트리이다.

컨텐츠-태그 테이블은 본 시스템에서 직접 사용되지는 않지만 XML을 분류 군집등을 하고자하는 애플리케이션에서 사용할 때 필요한 자료구조로써 구분하여 저장하였다. 본 시스템에서는 DTD태그부분만을 다루었지만, 후에 컨텐츠를 다루고자 할 때는 이를 이용하면 될 것이다. 정보 추출기의 과정은 [그림 3]과 같다.

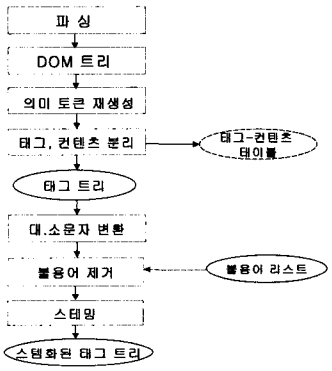


그림 3 정보 추출 과정

먼저 XML문서를 파서를 통해서 DOM트리를 생성한다. DOM트리로부터 본 논문에서 필요한 의미 있는 토큰을 재형성한다. 이때 의미 있는 토큰을 생성하기 위한 구별자로는 공백, 하이픈('-')과 언더스코어('_')를 사용하였다. 입력으로 하이픈과 언더스코어가 포함된 단어가 들어오면 하이픈과 언더스코어를 토큰화 과정에서 제거하여 각 토큰을 하나의 토큰으로 생성하였다. 하이픈과 언더스코어를 제거한 이유는 일반적으로 태그 작성 시

단어와 단어를 연결하고 싶을 때 하이픈, 언더스코어를 사용하기 때문이다. 따라서 하이픈과 언더스코어를 제거함으로써 각 단어의 의미를 모두 얻기 위함이다.

태그와 컨텐츠 분리단계에서는 의미 있는 토큰을 재형성하면서 태그와 컨텐츠를 구분하여 각각의 자료구조를 마련하여 각각 구분하여 생성하였다. 본 시스템에서 사용한 불용어 목록은 Brown corpus로부터 추출한 425개의 불용어 목록[4]에서 의미 있는 명사, 동사를 제외한 그 나머지를 사용하였다. 어간 추출 단계에서 용어로부터 어간(stem)을 추출함으로써, 복수형, 현재 진행형, 과거형 등으로 들어오는 단어의 원형을 찾는다. 따라서 이러한 단어를 가지고 WordNet과 사용자 정의 유사 용어 사전으로부터 해당 단어의 동의어를 찾는다. 본 논문에서는 어간 추출 알고리즘으로 Porter Suffix Stemmer[12]를 이용하였다.

3.4 동의어 벡터 생성기

정보 추출기로부터 생성된 스템화된 태그 트리의 정보를 입력으로 받아 각 태그를 최대한으로 동의어로 확장하기 위해서, 각 태그에 대한 동의어 벡터를 생성하는 단계이다. 동의어 벡터를 생성하기 위해 개념 지식을 사용하는 순서는 본 논문에서 직접 구축한 사용자 정의 유사 용어 사전을 먼저 살펴본 후에 WordNet을 사용하는 순서로 한다. 사용자 정의 유사 용어 사전을 먼저 살펴보는 이유는 다음과 같다. 태그로 사용자가 임의로 정의한 생략어가 WordNet에서 자연어로 존재한다면 다른 의미를 가질 수 있다. 예를 들면 <bibliography>를 <bib>로 표현한 경우에 <bib>라는 단어는 자연어로 '홀짝홀짝 마시다'라는 의미를 가지고 있다. 따라서 WordNet에서 <bib>라는 단어를 찾으면 그 동의어로써 <tipple>로 확장된다. 따라서 이러한 경우를 피하기 위해서 WordNet보다는 사용자 정의 유사 용어 사전을 먼저 검색한다. 사용자 정의 유사 용어 사전에서 검색하고자 하는 태그를 찾아서 동의어 벡터에 저장한다. 그 후 WordNet 데이타베이스를 연결하여 해당 태그에 대한 동의어 집합을 가져온다. 동의어 집합을 단어 단위로 쪼개어 각 동의어 단어를 해당 태그의 동의어 벡터에 저장한다. 다음 [그림 4]는 동의어 벡터 생성 과정이다.

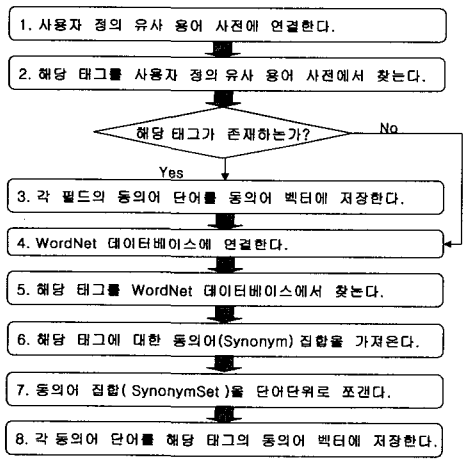


그림 4 동의어 벡터 생성 과정

3.5 유사도 측정기

유사도 측정 모듈에서는 이전 단계의 결과인 두 XML 문서의 확장된 태그를 입력으로 받아 두 문서의 태그간의 패턴을 의미적으로 비교한다. 그리고 기존의 문서모델인 벡터 스페이스 모델을 반구조적(semi-structured)문서인 XML문서에 확장 적용하여 두 XML태그간의 유사도를 구한다. 확장된 태그를 기반으로 두 XML문서의 태그를 비교하기 위한 기본 가정은 다음과 같다.

1. 두 문서로부터 추출된 태그의 용어가 완전히 일치하는 경우를 완전 매칭(exact matching)이라 한다.

2. 부분 매칭은 어떠한 태그 용어의 전체 스트링이 다른 태그에 완전히 포함되는 경우이다.

3. 완전히 일치하는 경우가 부분 매칭보다 우선 순위가 높다. (완전한 매칭(Exact) > 부분 매칭(Sub))

4. 동의어 벡터를 거치지 않고 직접 문서로부터 추출된 태그의 용어를 오리지널 태그(Ori)라 한다. 각 태그에 대해서 동의어 벡터를 통해 얻은 동의어들은 동의어 태그(Syn)라 한다.

5. 각 태그로부터 얻은 동의어 벡터내의 용어보다 오리지널 태그가 우선 순위가 높다.

(오리지널 태그의 용어 (Ori) > 동의어 벡터의 용어 (Syn))

다음 [표 2]는 두 XML문서의 확장된 각 태그의 패턴 매칭 레벨을 7 단계로 분류한 것이다.

표 2 태그의 의미적 유사도 레벨

비교 레벨	설 명	레벨 명칭
1	오리지널 태그간의 완전 일치	Ori-Ori Exact
2	오리지널 태그과 동의어 벡터의 용어간의 완전 일치	Ori-Syn Exact
3	동의어 벡터의 용어간의 완전 일치	Syn-Syn Exact
4	오리지널 태그간의 부분 일치	Ori-Ori Sub
5	오리지널 태그와 동의어 벡터의 용어간의 부분 일치	Ori-Syn Sub
6	동의어 벡터의 용어와 동의어 벡터의 용어간의 부분 일치	Syn-Syn Sub
7	두 태그간의 매칭 관계가 없을 때	이 외

다음은 두 XML문서의 태그간의 유사도를 측정하는 한 과정인 태그 유사행렬이다. 정보추출기, 동의어 벡터 생성기 등을 통해 나온 확장된 태그를 가지고 각 태그 간의 유사레벨을 파악한 것이다. 두 XML문서인 Books.xml과 Book_catalogue.xml을 대상으로 하였다.

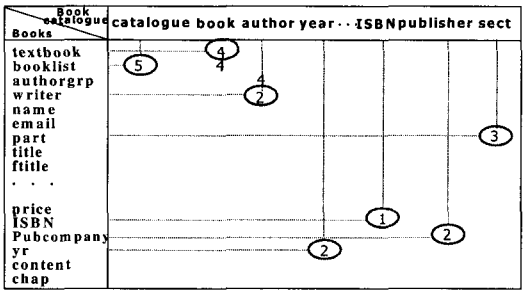


그림 5 태그 유사 행렬

3.5.1 벡터 스페이스 모델을 확장한 XML문서 모델

벡터 스페이스 모델은 정보 검색분야에서 문서를 표현하는 방법으로, 문서를 용어의 벡터로 표현한 것이다[5]. 즉 t개의 용어에 대해서 각 용어가 문서(Doc_i)에서 차지하는 가중치를 이용하여 벡터로 표현하였다.

두 문서의 유사도 계산은 용어가 각 문서에서 차지하는 가중치로 표현된 용어 벡터를 이용하여 계산한다. 이 때 각 용어가 문서에서 가지는 가중치를 계산하는 방법으로는 일반적으로 4가지 방법, 용어 빈도수(term frequency), 역문헌 빈도수(inverse document frequency), 용어 구별(term discrimination), 확률 인덱싱(probabilistic indexing)방법이 있다. 이중에서 용어 구별(term discrimination)은 문서를 구별 지울 수 있는 용어에 대해서 사용자가 용어 구별값(discrimination value)을 부여할 수 있는 방법이다. 단순히 용어의 빈도수만을 고려하여 가중치를 부여하는 방법과는 달리 용어 구별값을 이용한 가중치는 반구조적인 문서인 XML문서에서 태그가 가지는 중요도를 고려할 수 있는 방법이다. 따라서 태그의 의미적 유사도 레벨을 이용하여 가중치를 부여하였다.

문서 i에서 용어 k에 대한 가중치(WEIGHTik)는 용어 빈도수와 용어 구별값을 사용하여 다음과 같이 계산하였다.

$$WEIGHT_{ik} = FREQ_{ik} \cdot DISCVALUE_k \quad \text{(식1)}$$

- $FREQ_{ik}$ = 문서(i)에서 태그(k)의 발생회수
- $DISCVALUE_k$ = 태그(k)가 두 문서에서 가지는 의미 유사도 + 기준문서에서 태그(k)의 부모태그와 현문서에서의 부모태그와의 의미유사도

$DISCVALUE_k$의 파라미터 [태그(k)가 두 문서에서 가지는 의미 유사도]는 유사도 측정 모듈에서 태그의 유사도 행렬에서부터 나온 값으로써, 각 태그를 최대한 동의어 벡터로 확장해서 나온 단어들간의 유사도를 의미한다. 태그의 구조정보를 반영하기 위해서 [기준문서에서 태그(k)의 부모태그와 현 문서에서의 부모 태그와의 의미 유사도] 파라미터를 포함시켰다. 그 이유는 동일한 태그가 두 문서에서 나타났다하더라도 상위 태그의 의미에 따라 달라질 수 있기 때문이다. 예를 들면 두 문서에서 <name>이라는 태그가 동일하게 나타났다고 하더라도 상위태그가 <person>인 경우와 또 다른 문서에서 상위 태그가 <book>인 경우에는 동일한 <name>태그라 하더라도 의미상 차이가 있다. 따라서 본 논문에서는 비교하려는 태그의 상위 노드가 동일하다면 더욱 높은 가중치를 부여하였다.

위와 같은 방법으로 각 색인어가 각 문서에서 가지는 가중치를 계산하여 두 XML문서 각각을 벡터 모델로 표현하였다. 두 벡터에 대한 유사도는 몇 경우를 제외하고는 일반적으로 가장 많이 사용하는 코사인 상관계수(cosine coefficient)[4]를 이용하였다.

3.6 실험을 통한 시스템 검증

Tag Pattern Matcher 시스템의 타당성을 검증하기 위해 다음의 실험을 수행하였다. 실험 예제의 성격상 두 가지로 구분하여 수행하였다.

- 실험타입1 : 동일한 내용을 상이한 태그와 구조로 표현한 경우
- 실험타입2 : 상이한 내용과 태그로 표현하였지만 비슷한 부류로 판단될 수 있는 경우

실험타입1의 방법 및 결과를 간단히 설명하면 다음과 같다. 실험으로 사용한 예제는 온라인 책 서점인 Amazon 사이트와 Barnes&Noble 사이트에서 "professional WAP" 책을 검색하여 나온 결과 페이지의 내용을 참조하여 이를 XML로 문서화한 것이다. 각 문서에 나타난 태그 구조는 다음과 같다.

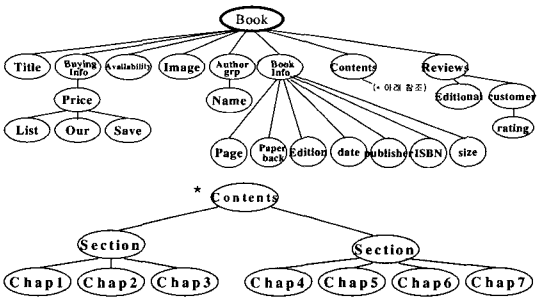


그림 6 실험1예제: WAP_Amazon.xml의 태그구조

다음은 WAP_B&N.xml 문서와 태그 구조이다.

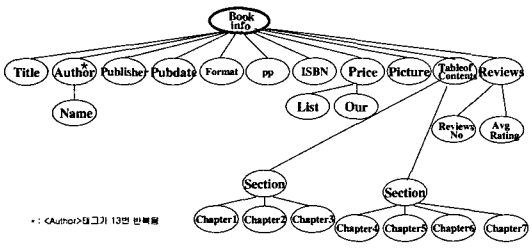


그림 7 실험 예제 1 : WAP_B&N.xml 문서의 태그구조

① 기존의 벡터 스페이스 모델 적용

벡터 스페이스 모델을 이용하여 두 문서의 유사도란 기준문서에 대해서 다른 문서가 어느 정도 유사한지를 의미한다. 따라서 본 예제에서 기준문서는 WAP_Amazon.xml문서를 기준으로 하였다. 따라서 인덱스는 WAP_Amazon.xml문서에 나타나는 모든 태그를 인덱스 집합으로 하였다. 각 인덱스가 WAP_Amazon.xml문서에서 발생하는 빈도수를 가중치로 부여하여 WAP_Amazon.xml문서를 벡터 모델로 표현하였고, WAP_B&N.xml에서 발생하는 빈도수(frequency)를 가중치로 부여하여 WAP_B&N.xml문서를 벡터 모델로 표현하여, 두 벡터간의 코사인 값을 계산하였다. 다음은 각 문서에 대한 벡터모델이다. 편의상 WAP_amaxon.xml을 X, WAP_b&n.xml를 Y라고 하자. x_i는 X에서 각 인덱스(i)에 대한 가중치이고, y_i는 문서B에서 각 인덱스(i)에 대한 가중치이다. T는 인덱스의 집합이고, t는 인덱스의 개수이다.

② 본 논문에서 제시하는 방법

본 논문에서 제시하는 방법인 태그의 의미적 정보를 고려한 가중치를 3.5.1절에서 설명한 식에 의해서 계산하였다. 다음은 태그의 의미적 정보를 고려한 벡터 스페이스 모델에 따라 표현된 두 문서의 벡터이다.

위의 두 벡터간의 코싸인 유사도를 구한 결과, 두 문서의 유사도를 기존의 벡터 스페이스 모델을 적용했을

T =(Book, Title, BuyingInfo, Price, List, Our, Save, Availability, Image, Aughorgsrpp, Name, BookInfo, Page, Paperback, Edition, Date, Publisher, ISBN, Size, Contents, Section, Chap1, Chap2, Chap3, Chap4, Chap5, Chap6, Chap7, Reviews, Editional, Customer, Rating)
X = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
Y = (0, 1, 0, 1, 0, 1, 1, 0, 0, 0,13, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0)

그림 8 기존의 벡터스페이스모델을 이용한 벡터 표현

T = (Book, Title, BuyingInfo, Price, List, Our, Save, Availability, Image, Aughorgrpp, Name, BookInfo, Page, Paperback, Edition, Date, Publisher, ISBN, Size, Contents, Section, Chap1, Chap2, Chap3, Chap4, Chap5, Chap6, Chap7, Reciews, Editional, Customer, Rating)
X = (12, 12, 12, 18, 24, 24, 24, 12, 12, 12, 234, 12, 18, 18, 18, 18, 18, 18, 12, 36, 24, 24, 24, 24, 24, 24, 24, 12, 18, 18, 24)
Y = (6, 9, 0, 7, 0, 12, 8, 8, 8, 78, 117, 9, 11, 0, 0, 9, 12, 12, 0, 8, 22, 11, 11, 11, 11, 11, 11, 11, 9, 0, 0, 3)

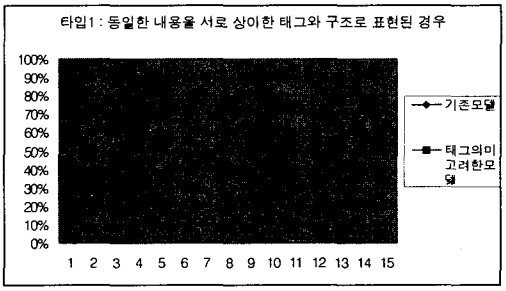
그림 9 태그의 의미적 정보를 고려한 벡터 표현

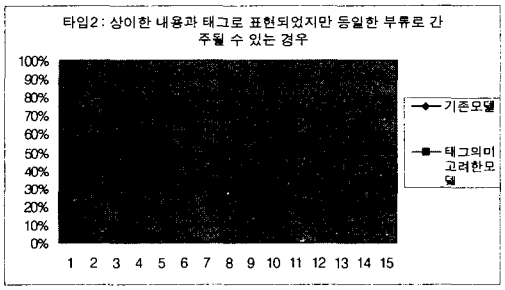
때와 본 논문에서 제시한 Tag Pattern Matcher를 통하였을 때 유사도는 다음과 같다. Tag Pattern Matcher를 통한 유사도 계산은 기존의 벡터스페이스 모델을 적용했을 경우의 유사도인 30.2%보다 2.83배 높은 유사도를 나타낸다.

표 3 유사도 비교

	기존 모델	Tag Pattern Matcher
두 문서의 유사도	30.2 (%)	85.64 (%)

각 실험 타입별로 실험한 결과는 다음과 같다.





이러한 결과는 두 문서가 가지는 태그의 의미를 고려하여 각 태그에 대한 동의어를 이끌어내었기 때문이다. 실험타입2 경우는 실제로 인간이 휴리스틱하게 두 문서

를 비교하였을 때도 본 논문에서 제시한 모델을 적용하였을 때의 결과와 비슷한 결과를 얻을 수 있다.

4. 결론 및 향후 연구

본 논문에서는 XML 태그가 지니고 있는 의미 정보(semantic information)를 추출하여 각 태그 용어에 대해서 최대한 의미적으로 유사한 단어 혹은 개념으로 확장하여 벡터 모델로 XML태그를 표현함으로써, 두 XML태그 간의 유사도를 측정하였다. 그 결과 두 XML문서가 유사한지 아닌지를 인간이 휴리스틱하게 판단할 수 있듯이, 본 논문에서 제시한 방법을 통해 XML태그의 유사도를 파악 할 수 있었다.

본 논문의 의의는 크게 두 가지로 다음과 같다.

첫째, 정보 검색분야에서 문서를 표현하는 방법으로 가장 널리 사용되는 벡터 스페이스 모델을 반구조적(semi-structured) 문서인 XML에 적용하였다는 점이다. 따라서 XML만의 고유한 특징인 태그가 지니고 있는 의미 정보를 반영하였다.

둘째, XML문서를 분류하거나 XML문서를 대상으로 비슷한 문서끼리 마이닝하는 응용분야에서 본 논문에서 제시하는 개념 기반의 태그 패턴 매칭 방법을 데이타 준비 작업으로 사용할 수 있다. 본 논문에서 XML문서에서 태그와 내용(content)를 분리하였고 추출된 태그에서 불용어(stoplist)를 제거하고 어간 추출(stemming) 과정인 전처리 작업을 수행하였다. 이러한 작업 후에 시소러스를 통하여 최대한 유사한 태그의 노드 정보를 밝혀내었다. 이 결과를 이용하면 XML문서를 분류, 군집등의 마이닝을 위한 데이타 준비(data preparation)과정으로 사용될 수 있다.

향후 연구 과제는 보다 정확한 구조 정보를 추출하고 XML문서의 태그뿐 아니라, 컨텐츠 모두를 고려하여 XML문서의 유사성을 파악하는 것이다. 물론 본 논문에서 XML문서의 구조 정보를 반영하기 위해 태그의 부모 노드 태그의 유사도 정도를 고려하였지만 문서 전체의 보다 정확한 구조정보를 파악해야할 것이다.

참 고 문 헌

[1] Minnos N. Garofalakis, Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim, "Of Crawlers, Portals, Mice, and Men : Is there more to Mining the Web?," In *Proc. of the ACM SIGMOD Int. Conf. Management of Data*, pages 504, Philadelphia, PA, USA, 1999.

[2] William J. Pardi, *XML in Action*, Microsoft Press, 1999.

[3] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. *Extensible Markup Language (XML) 1.0, W3C Recommendation.* World Wide Web Consortium, Feb. 1998. http://www.w3.org/TR/1998/REC-xml-19980210.

[4] William B. Frakes and Ricardo Baeza-Yates, *Information Retrieval: Data Structures & Algorithms*, London:Prentice Hall, 1995.

[5] Gerard Salton and Michael J. McGill, *Introduction to Modern Information Retrieval,* McGraw-Hill Book Company, New York, 1983.

[6] 황도삼, 최기선, 김태석 공역, 자연언어 처리, 홍릉과학출판사, 1998.

[7] Fellbaum, C. 1998. *Wordnet: An Electronic Lexical Database. Cambridge:*MIT Press.

[8] Miller G.A, Beckwith R., Fellbaum C., Gross D. and Miller K., "Intorduction to WordNet : An On-Line Lexical Database." in Five Papers on WordNet, CSL report, Cognitive Science Laboratory, Princeton University, 1993.

[9] R.Richardson, A.F.Smeaton, and J.Murphy, "Using WordNet as a Knowledge Base for Measuring Semantic similarity between Words," Working Paper:CA-1294.

[10] David Megginson, *Structuring XML Documents,* Prentice Hall PTR, 1998.

[11] Norman Walsh and Leonard Muellner, *DocBook: The Definitive Guide*, O'REILLY, 1999.

[12] M.Porter. An Algorithm for suffix stripping. Program, 14(3), pages130-137, 1980.

이 혜 수

1994년 3월 ~ 1998년 2월 숙명여대 전자계산학과 학사. 1998년 3월 ~ 1999년 2월 LG 히다찌(S/W 개발) 사원. 1999년 3월 ~ 2001년 2월 이화여대 컴퓨터학과 석사. 2001년 1월 ~ 현재 삼성전자 연구원(S/W Center). 관심분야는 XML, Embedded Programming

이 정 원

정보과학회논문지 : 소프트웨어 및 응용 제 29 권 제 6 호 참조

이 기 호

정보과학회논문지 : 소프트웨어 및 응용 제 29 권 제 6 호 참조