XML 필터링을 위한 WFilter(Weighted Filter)

최정필°, 최오훈, 백두권
고려대학교 컴퓨터학과 소프트웨어 시스템 연구실
(studriver°, pens, baik)@software.korea.ac.kr

WFilter (Weighted Filter) for XML filtering

Jung-Pil Choi°, O-Hoon Choi, Doo-Kwon Baik
Software System Lab. Dept of Computer Science & Engineering, Korea University

요 약

XML 문서를 비롯하여 인터넷을 통해 교환되는 문서의 비약적인 증가로 인하여, 불필요한 문서에 대한 필터링 및 문서 내의 데이터를 필터링하여 정보를 선택적으로 사용하고자 하는 사용자의 요구가 증대되었다. 기존 XML 필터링 방식은 질의 구조에 의존적이기 때문에, 질의 증가에 따른 필터링 인덱스 구성 및 유지의 문제점을 야기할 수 있다. 본 논문에서는 정보 추출 분야에서 널리 사용되는 단어 벡터의 개념을 사용하여 선택적으로 질의에 가중치를 주어 데이터를 효율적으로 추출할 수 있는 XML WFilter (Weighted Filtering) 기법을 제안한다.

1. 서 론

인터넷의 발달이 이루어짐에 따라 일반 사용자도 손쉽게 방대한 양의 정보를 접할 수 있는 환경이 조성되었다. 이러한 추세에 부합하여, 사용자의 요구에 맞추어 필요한 정보를 선택적으로 공급하고자 하는 목적으로 Selective Dissemination of Information(SDI)과 같은 시스템이 개발되고 있다.[1] XML이 인터넷 상의 정보 교환 표준으로 대두됨에 따라 웹 상의 많은 정보들이 XML 형식으로 표현되고 있으며, 이러한 XML 문서들을 필터링 해주는 XML Filtering 기술도 함께 연구중이다.

기존의 연구에서는 XML 문서의 스키마 구조를 기반으로 XML 경로 지정 문법인 XPath 표현에 따라 필터링을 구현하는 XTrie[2], YFilter[3]와 같은 시스템이 설계되었다. 그러나 이러한 필터링 기법들은 사용자 프로파일이 증가함에 따라 인덱스도 함께 성장하여 거대해진 인덱스의 관리 문제가 발생할 수 있다. 또한, 단순히 XPath 표현식에 맞는 문서를 추출하는 방식으로는, 사용자가 원하는 내용에 관련된 XML 문서 내의 데이터를 선택적으로 추출하여 사용자에게 공급하고자 하는 의도와는 다른 결과를 초래할 수 있다.

본 논문에서는 이러한 문제점을 개선하기 위하여 정보 추출 분야에서 널리 사용되는 단어 벡터의 개념을 이용하여 선택적으로 데이터를 추출할 수 있는 XML WFilter (Weighted Filter) 기법을 제안한다. WFilter는 질의 분석시 가중치를 부가하여 인덱스 탐색 효율을 높이는 방법이다.

본 논문의 구성은 다음과 같다. 2장은 SDI 시스템 및 기존의 필터링 방법에 대해서 소개하고 문제점에 대해서 이야기한다. 3장에서 본 논문에서 제시된 기법에 대해서 설명한 후, 마지막으로 4장에서는 결론 및 향후 연구에 대해서 논의한다.

2. 관련 연구

2.1 XML 기반 SDI

SDI는 다수의 사용자를 대상으로 최신의 데이터를 배포하는 서비스이다. 시스템에 새로운 데이터가 발생하면, 이를 사용자의 관심 분야를 표현한 사용자 프로파일을 기준으로 필터링한 뒤 해당 데이터만을 배포한다.[1] XML이 인터넷 상의 데이터 교환 표준으로 등장하면서 이러한 XML 문서에 대한 SDI 시스템이 연구되었다. 기존의 SDI 시스템에서는 일반적인 키워드 검색을 위한 사용자 프로파일을 구성한데 반해, XML 기반 SDI에서는 XML 문서의 구조적 특성을 반영하기 위해 XPath를 이용하여 작성되었다. XPath는 XML 문서 상의 경로를 지정하기 위한 문법이다.[4]

사용자에 의한 직접 입력 혹은 사용자의 접근 패턴에 근거한 기계 학습에 의해 사용자 프로파일이 작성되면, 이는 XPath 형식으로 인덱스로 구성된다. 새롭게 작성되거나 변환된 XML 문서가 입력되면, 이 데이터를 전송할 사용자를 결정하기 위해 사용자 프로파일을 근거로 필터링을 수행한다.

2.2 XML 필터링

기존의 필터링 기법들은 사용자 프로파일인 질의를 처리하기 위하여 트리구조, FSM 구조 등을 구성하여 XML 문서를 필터링 하여 처리하였다. 또한 기존의 필터링 기법들은 사용자 프로파일 XPath의 인덱싱 구조를 개선함으로써 효율적인 필터링 기법을 제안하고 있

다. YFilter[2]의 예를 보면, 각 XPath 질의들을 하나의 비결정적 유한 오토마타(NFA : Non-deterministic Finite Automata)로 구성하여, 해당 XML 문서를 파싱하는 과정에서 각 질의가 맵핑된 최종 상태(final state)에 도달하는지의 여부로 문서들을 필터링한다.

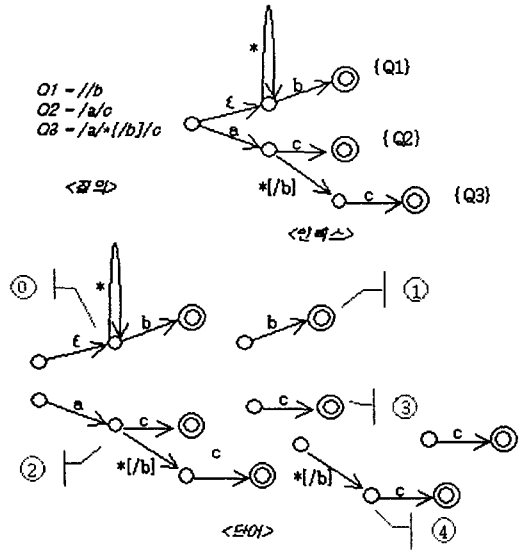
그러나 이러한 기법들은 공통적으로 인덱싱 구조의 종류[1,2,3]에 관계없이, 사용자의 증가에 따라 인덱스도 함께 규모가 커지는 문제점을 내포하고 있다. 즉, 사용자의 증가는 사용자 프로파일의 증가를 유발하고, 이에 따라 인덱스를 탐색하는 오버헤드도 증가하게 된다. 이러한 문제점은 인터넷과 같은 대규모 분산 환경에서 심각하게 고려될 수 있다. 또한, 기존의 기법들은 XPath로 표현된 사용자 프로파일의 구조가 해당 XML 문서의 구조내에 포함되어 있는지의 여부로 필터링을 하기 때문에 충분한 선택성을 갖지 못한다.

본 논문에서는 정보 추출 분야에서 널리 사용되는 단어 벡터의 개념을 사용하여 선택적으로 데이터를 추출할 수 있는 필터링 기술을 기존 YFilter[3]의 NFA에 접목한 Weighted Filter (WFilter) 기법을 제안한다.

3. WFilter (Weighted Filter)

3.1 YFilter에 기반한 weighted NFA

YFilter의 기본적인 원리는 주어진 질의로 NFA 인덱스를 구성하고, XML 문서의 요소(element)에 따라서 상태 전이를 하는 것이다.[3] YFilter[3]에서와는 달리 인덱스 탐색시 XML 문서의 질의 유사성에 기반한 선택성을 부여하기 위해, 인덱스 내의 각 노드 간 경로에 가중치를 부여한다. 즉, XML 문서내의 단어 빈도에 따라 상태 전이를 한다. 본 논문에서는 YFilter[3]에서 사용된 NFA 인덱스 구조를 바탕으로 설명을 하지만, XTrie[2]와 같은 Trie 구조 및 다른 인덱스 구조에도 응용 가능하다.



[그림 1] 질의 집합으로부터 생성된 인덱스와 이로부터 추출된 단어들

문서내의 단어 빈도에 따른 경로의 가중치를 부여하기 위하여 벡터 공간 모델(VSM : Vector Space Model)[5]을 사용한다 벡터 공간 모델에서는 문서와 질의를 단어 벡터(term vector)로 표현한다. 일반적인 벡터 공간 모델을 확장하여, 인덱스를 구성하는 각 노드에서 그 노드를 루트로 하는 각각의 서브트리를 단어(term)로 간주한다.[6]

[그림 1]은 몇 개의 질의로 구성된 NFA 인덱스 구조와 이로부터 추출된 단어들을 보여주고 있다. 그림을 살펴보면 우선 전체 트리를 루트를 중심으로 두개의 서브트리인 ⓪과 ②로 나눌 수 있으며, 각각이 하나의 단어가 된다. 단 ⓪은 descendant operator('//')가 포함되었기 때문에 2개의 단어로 분리하며, 여기서는 ①과 중복된다. 서브트리 ②는 다시 서브트리 ③과 ④로 나뉘어진다. 결국 위 인덱스로부터 ①~④의 단어를 추출하였다.

각 단어에 대하여 단어 벡터를 계산하는데, 이 단어 벡터는 해당 질의 및 문서에서 그 단어가 갖는 중요성을 나타내며 그 서브트리로의 경로에 대한 가중치가 된다. 단어 벡터는 다음과 같이 계산된다.

$$Term\ Vector = TF * IDF$$

(TF : Term Frequency, IDF : Inverse Document Frequency)

단, 하부에 다시 서브트리를 갖는 단어의 경우는, 먼저 서브트리들의 단어 벡터를 계산하고 그 값들에 대한 함수값을 전체 단어의 단어 벡터로 한다.[6]

한 단어가 한 문서 내에 높은 빈도로 존재한다면, 그것은 그 단어가 그 문서를 다른 문서들과 차별화해줄 수 있는 단어가 될 수 있다는 뜻이다. 반면 그 단어가 다른 문서들에서도 많이 나타난다면, 그 단어의 중요성은 감소된다.

단어별 IDF에 대한 테이블을 유지하고 있다가, 새로운 XML 문서가 입력되면 이를 바탕으로 단어 벡터를 실시간으로 계산한다. [그림 2]는 이를 위한 테이블 구조의 예를 보여준다. DF(Document Frequency)는 해당 단어의 다른 문서에서의 발생 빈도를 저장하고 있다. 계산시에는 DF의 역인 IDF를 취해서 TF와의 곱으로 계산한다.

단어	DF(IDF)	단어 벡터
①	366	0.0081
②	123	0.0002
③	435	0.0065
④	211	0.0015

[그림 2] 단어 벡터 테이블

3.2 WFilter를 사용한 XML 문서 파싱 및 필터링

새로운 XML문서가 입력되면 단어 별 단어 출현 빈도를 계산하기 위해 첫번째 파싱을 수행한다. 파싱 과정에서 테이블 상의 각 단어별로 TF를 구할 수 있으며, 이를 바탕으로 다시 단어 벡터를 계산한다. 이 단어 벡터는 단어 벡터 테이블에 저장되어, 입력된 문서의 단어 분포 특성을 표현한다.

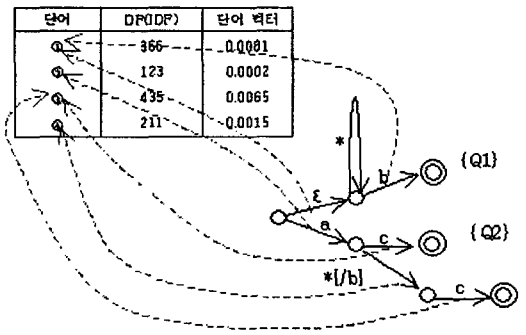
두번째 파싱 과정에서 실제 필터링이 이루어진다. 최초의 NFA는 초기 상태(initial state)에 있다가 파싱이 진행됨에 따라 전이한다. 파싱은 event-driven 방식으로, XML문서 상에서 순차적으로 진행하다가 "start-of-element(요소 시작)"이나 "end-of-element(요소 끝)" 등의 이벤트가 발생하면 NFA에서 전이가 이루어진다.

Start-of-element : 파서가 어떤 요소의 시작 태그를 만나면 그 요소의 명칭을 알려준다. NFA 상의 현재 후보 노드 중에서 그 요소 명칭을 통해 전이할 수 있는 경로를 탐색한다. 전이 가능한 노드가 최종 상태(final state)인 경우에 해당 질의가 필터링을 통과한 것으로 간주한다. 그렇지 않은 경우, 다음 단계의 후보 노드에 추가시켜서 다음 단계의 전이가 가능하도록 한다.

End-of-element : 요소의 끝 태그를 만났다면, 대응 시작 태그를 만나기 이전의 상태로 되돌리기 위해 해당 후보 노드를 삭제한다

3.3 상태 전이(transition)의 제한

결국 WFilter의 weighted NFA는 노드를 모두 탐색하지 않게 하도록 상태 전이를 제한하기 위한 것이다. 상태 전이를 제한하기 위한 후보 노드의 관리 방법에 따라 크게 두가지 용도로 활용될 수 있다.



[그림 3] WFilter : weighted filter

첫째, 질의와 연관성이 적은 문서들을 필터링 대상에서 제외시킨다. 후보 노드로의 경로가 갖는 단어 벡터는 필터링 대상 문서와 해당 질의 간의 연관성을 나타내는 요소이다. 특정 임계점을 지정하여 이 임계점보다 단어 벡터 값이 적은 노드를 다음 단계 후보 노드에서 제외시킴으로써, 벡터 공간 모델 관점에서 벡터 값이 높은 단어를 포함한 질의만으로 후보 노드를 제한할 수 있다.

둘째, 후보 노드 집합의 크기를 제한시켜 NFA의 상태 전이가 더 이상 확장되지 않도록 막는다. 사용자 프로파일이 복잡해지거나 그 수가 늘어날수록 NFA의 전체 노드 수도 증가하게 된다. 이에 후보 노드를 단어 벡터 순으로 우선 순위를 주고 후보 노드 집합의 허용 범위 내에 있는 후보 노드만을 필터링 대상으로 한다. 즉, 시스템의 동작에 있어 바람직한 수준의 후보 노드 집합 한계를 설정하여 이를 초과하는 만큼의 후보 노드는 단어 벡터 값이 적은 것을 우선으로 하여 필터링에서 제외시킨다.

[그림 3]은 NFA의 각 경로마다 해당 단어에 해당하는 단어 벡터 테이블의 참조를 두어 weighted NFA를 구성한 모습을 나타낸다. 위에서 설명한 두 가지 경우와 같이 상태 전이를 제한 시킬 경우, 각 경로의 단어 벡터를 참조하여 결정한다. 예를 들어, '/a'에서 '/c'와 '/*[/b]/c'의 두 상태를 모두 후보 상태라고 했을 때, 단어 벡터 테이블의 ③과 ④를 비교하여 결정한다.

4. 결론 및 향후 연구

기존의 XML 필터링 기법들은 질의 구조에 의존적이기 때문에, 질의 자체가 완벽하게 본래의 의도를 반영하지 못하거나 질의 문법 자체의 한계로 인해서 효과적으로 의미 있는 결과를 이끌어내지 못한다. 또, 주어진 모든 질의에 대해서 완전한 결과 집단을 도출하는 방식을 사용하고 있기 때문에, 질의의 수가 급속히 증가하는 경우 오버헤드를 줄이기 위한 관리가 요구된다. 본 논문에서는 필터링을 선택적으로 수행하기 위한 수단으로, 정보 추출 분야에서 널리 사용되는 단어 벡터의 개념을 도입한 WFilter를 제안하여 이러한 문제점의 해결을 모색하였다. 본 논문에서 제안한 WFilter는 필터링 시 가중치를 부여한 인덱스 구조를 형성하여 정보추출의 정확성을 높였다.

향후에는 XML이라는 환경에서 단어(term)가 갖는 의미에 따른 단어 벡터 산출 및 적용 방식의 최적화에 대한 연구, 기존 필터링 방식의 개선된 기법 연구와 더불어, XML 문서의 의미 기반 필터링에 대한 연구가 이루어지게 될 것이다.

5. 참고문헌

[1] Altinel, M., Franklin, M. J., Efficient filtering of XML documents for selective dissemination of information, In Proceedings of VLDB Conference (2000)

[2] Chan, C. Y., Felber, P., Garofalakis, M. N., Rastogi, R., Efficient filtering of XML documents with XPath expressions, In Proceedings of IEEE Conference on Data Engineering (2003)

[3] Yanlei Diao, Michael J. Franklin, High-Performance XML Filtering: An Overview of YFilter, IEEE Data Engineering Bulletin 26(1): 41-48 (2003)

[4] Clark, J., DeRose, S, XML path language XPath - version 1.0, http://www.w3g.org/TR/path, Novermber 1999

[5] G. Salton, M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Tokio, 1983

[6] T. Schlieder, H. Meuss, Result ranking for structured queries against XML documents, In DELOS Workshop on Information Seeking, Searching and Querying in Digital Libraries, Zurich, Switzerland, December 2000