컨텐츠 메타데이터 통합 수집 장치에서의 중복 컨텐츠 필터링 기능 구현

조상욱° 이민호

삼성전자 디지털미디어연구소

swkhan.cho@samsung.com, minho03.lee@samsung.com

Filtering function embodiment of duplicated contents in integrated apparatus of content metadata aggregation

Sangwook Cho° Minho Lee

Samsung Electronics, Digital Media R&D Center

요 약

무한 웹 컨텐츠 환경에서는 사용자의 컨텐츠 선택을 용이하게 하기 위하여 메타데이터를 다양한 방법으로 수집할 수 있다. 그러나 한 가지 방법으로는 메타데이터의 수신이 제한적이고 풍부한 메타데이터 수신을 위해서는 다양한 방법을 이용해야 한다. 그래서 본 논문에서는 메타데이터 수집 방법들을 통합하는 장치를 제안하고, 통합 메타데이터의 품질 향상을 위해 통합과정에서 발생하는 중복 메타데이터의 필터링 방법을 제시 및 검증한다. 구체적으로는 현재 웹 상에서 다양하게 제공되고 있는 메타데이터 수집 기능들을 분석하고, 통합 장치의 개념적인 구조를 제시하며, 웹 상에서 많이 보급되고 있는 RSS Reader를 통해 메타데이터를 수집하고 이를 토대로 분석하여 중복 컨텐츠를 판단하는 방법을 제안하였다.

1. 서 론

웹 접속 환경의 발달과 IPTV의 보급으로 웹으로부터 제공되는 컨텐츠의 수는 기하급수적으로 증가하고 있다. 이러한 상황에서 사용자는 컨텐츠 선택에 어려움이 발생하게 되었다. 그래서 웹 환경에서는 사용자가 원하는 컨텐츠를 빠르고 정확하게 찾도록 구조화된 메타데이터를 제공하고 있고[1], 그 방법은 다양한 형태로 이루어지고 있다. 또한, 컨텐츠의 제공 및 선택 수준 향상을 위해서는 메타데이터를 풍부하게 가지는 것이 중요하다. 그렇지만 개별의 메타데이터 수신 기술은 범위가 제한적이기 때문에 웹에 존재하는 메타데이터의 작은 부분만을 이용할 뿐이다. 또한 개별 기술 별로 수신 메타데이터가 다르기 때문에 메타데이터의 활용에 어려움이 따른다.

본 논문에서는 메타데이터를 이용하고자 하는 응용 프로그램이 다양한 방식으로 서비스 되고 있는 메타데이터 제공 형태를 모두 지원할 수 있도록 메타데이터 통합 수집 장치를 제안하고, 통합 과정에서 발생하는 중복 메타데이터를 확인하며 이를 필터링 하는 방법을 제안한다. 이러한 통합 장치는 풍부한 통합 메타데이터를 생성하고, 필터링을 통해 통합 메타데이터의 품질 향상을 도모하여 응용시스템의 효율을 높일 수 있다. 이를 위해 구체적으로 2장에서 다양한 메타데이터 제공방법을 설명하고, 3장에서 메타데이터 제공 기술들의 통합 방법에 대한 개념을 제안한다. 그리고 4장에서 통합 과정 중 발생하는 중복 메타데이터를 확인하고, 5장에서 중복 메타데이터를 필터링하는 방법을 제시한 후, 실제 적용 결과를 확인하였다.

2. 다양한 메타데이터 제공 방법

메타데이터는 하나의 컨텐츠 데이터에 대한 부가 정보를 의미하고, 메타데이터 수신 기술은 컨텐츠와 관련된 정보를 웹으로부터 얻어 올 수 있도록 하는 기술로 정의한다. 현재 웹 상에서는 다양하게 메타데이터를 수신하는 방법이 제안되고 시스템이 구축되고 있고, SD&S(Service Discovery & Selection)기반 EPG기술[2], OpenAPI를 이용하는 기술[3], Syndication Format (RSS/ATOM)을 이용한 기술 등이 실제 적용되고 있거나 적용될 예정이다[4].

SD&S EPG데이터는 IPTV에서 채널 검색 및 선택에 관하여 정의된 메타데이터이다. EPG데이터 송수신을 위해 Server와 Client가 존재하고 정의된 메타데이터 구조에 따라 통신을 한다. 메타데이터 수신 측에서는 SD&S Client 역할을 하는 모듈을 통해 SD&S Service를 제공하는 Server로부터 메타데이터(EPG)를 수신한다.

OpenAPI는 웹 사이트들이 제공하는 기능의 일부를 API로 공개한 것이다. 많은 사이트들이 검색과 지도 서비스 위주로 OpenAPI를 제공하고 있으며 HTTP

프로토콜을 이용한다. GET 이나 POST와 같은 요청을 서비스 제공 URL로 보내고 XML로 구성된 응답 문서를 받는다. 그러나, OpenAPI의 응답 XML 문서 형식은 서비스 제공 사이트마다 각기 다르다.

Syndication Format[4]을 이용하는 방법은 RSS/ATOM 규격을 이용하는 것이다. Format의 사용 목적은 동일하나 제정 표준에 따라 규격이 RSS 1.0, RSS 2.0, ATOM으로 나뉜다. 세부적인 내용의 차이는 있으나 XML을 바탕으로 규격이 구성되어 있다. 그래서 앞으로 본 논문에서는 Syndication Format 규격을 별도로 구분하지 않고 RSS로 통칭한다. Syndication Format도 OpenAPI와 같이 HTTP 프로토콜을 사용한다. 미리 등록된 RSS 서비스 제공 주소로 규격 문서를 주기적으로 요청하여 받아 온다. 메타데이터를 받아오기 위해 미리 등록된 RSS 주소리스트는 Feed목록 이라 불리며, 주기적인 수신을 위해서 Feed목록에 Feed주소들은 저장되어 있어야 한다.

각각의 기술들은 웹 환경에서 메타데이터를 수신할 수 있는 다양한 방법을 제공하고 있으나, 메타데이터는 수신 기술마다 다르다는 단점이 있다. 그래서 각 기술을 동시에 이용하여 응용 프로그램이 사용 가능한 통합 메타데이터를 생성하는 경우에는 통합을 위한 추가적인 장치가 필요하다.

3. 메타데이터 수집 기능의 통합 방법

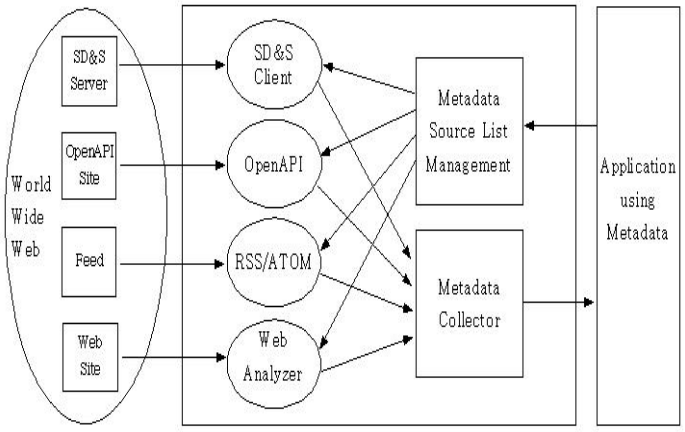

그림 1. 제안된 메타데이터 통합 시스템 개념도

웹 상에서는 다양한 방식으로 메타데이터를 제공하는 기술이 발전하고 있으며, 이러한 기술들을 바탕으로 다양한 서비스를 창출하고자 하는 연구도 함께 진행되고 있다. 이와 같은 서비스와 연구 분야에는 데이터 사용의 편의와 효율성, 호환을 위해서 주로 XML을 이용하고 있고, 현재까지 제시된 웹 상의 메타데이터 제공 기술들도 XML을 기본으로 하고 있다[5]. 따라서 통합을 위한 장치에서도 XML을 이용한 통합 메타데이터를 생성한다.

구체적인 시스템의 개념도는 그림1과 같다. 전체 시스템의 구성은 메타데이터 통신 기술들의 수신 Client기능과 각 기술의 Metadata를 통합하는 Metadata Collector, Metadata를 제공하는 웹 주소들을 관리하는 Metadata Source List Management로 이뤄져 있다. 그리고, 실제적인 통합을 수행하는 Metadata Collector는 그림 2와 같이 Metadata수신부, XML 파싱부, Data처리부, 중복 제거부로 구성된다. Metadata수신부 에서 각 기술별 메타데이터를 받아오고, 수집된 XML형태의 메타데이터는 XML 처리부를 통해 분석되며 Data 처리부를 거치면서 메타데이터에서 키워드를 추출하고 저장한다. 마지막으로 중복 제거부를 통해 필터링 과정을 수행하여 통합 메타데이터를 생성한다. 중복 여부의 판단은 자연언어 처리된 키워드를 이용하여 키워드를 비교하는 방식을 사용하였다.

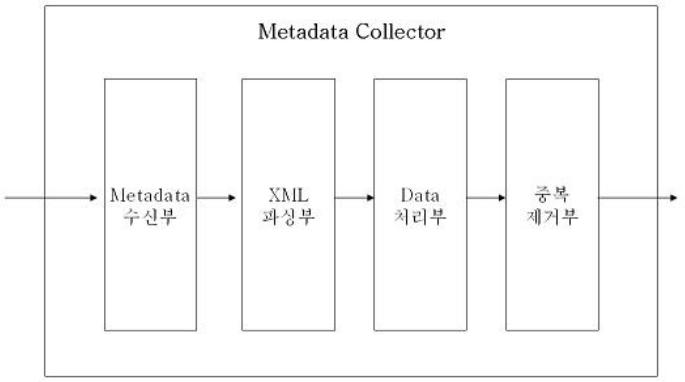

그림 2. Metadata Collector 세부 개념도

제공되는 웹 컨텐츠중 Metadata를 제공하는 컨텐츠는 아직까지 많지 않다. 그래서 HTML로 구성된 많은 양의 웹 자원들도 Metadata로 이용이 가능할 수 있도록 해야 한다. 이를 위해 일반적인 HTML 웹 사이트를 분석하여 Metadata를 생성할 수 있는 장치인 Web Analyzer가 필요하다. 검색 엔진에서 주로 사용하는 Web Crawler[6]와 같이 웹 페이지에서 키워드를 추출하고, 추가로 메타데이터를 생성하는 기능이 들어있는 Web Analyzer도 통합 장치에 포함된다.

메타데이터 통합을 위해서는 메타데이터 응답 문서들을 분석하고 통합하는 과정이 필요하다. 통합 과정은 메타데이터 포맷을 정해진 규격에 맞게 일괄적으로 수정하는 기능이 포함하여야 하고 이 과정은 Metadata Collector에서 수행한다. 통합된 메타데이터 저장 포맷으로 XML을 사용하며, XML Schema는 통합 메타데이터를 사용하는 응용

프로그램별로 다른 정의가 가능하도록 해야 한다. 또한 Metadata Source List Management부에서는 각 기술 서비스를 제공하는 URL 리스트를 저장하고, 메타데이터 수신 시 연관URL을 선택한다. 메타데이터를 받아오는 웹 사이트는 유동적이기 때문에 사이트를 추가하거라 삭제하는 기능도 필요하다.

4. 통합 메타데이터 중복 확인 실험

다양한 메타데이터 수신 기술로부터 통합 메타데이터를 생성하는 과정에서는 중복된 데이터가 발생할 가능성이 높다. 구체적인 예를 들자면 현재 OpenAPI를 서비스하는 Naver[7]에서 OpenAPI를 이용하여 뉴스 검색을 하는 것과 Naver의 뉴스 Feed를 등록하여 RSS를 받는 경우엔 같은 내용을 받을 가능성이 높다. 또한 RSS만 이용하여 수신 하여도 중복 컨텐츠의 메타데이터를 받는 일이 빈번하게 발생한다. 다양한 Feed에서 RSS 데이터를 수신하지만, 각 Feed를 통해 모두 자신만의 독창적인 컨텐츠를 제공하는 것은 아니고, 많은 사용자들이 스크랩을 하거나 같은 내용을 인용해 쓰는 것이 그 이유이다.[8] 그래서 메타데이터의 중복여부를 확인하고 중복 비율을 측정하고자 실제 응용 프로그램을 통한 실험을 설계하고 수행하였다. 전체 통합 시스템을 구축하지 않아도 Syndication Format을 이용한 메타데이터 수신만으로 중복 확인이 가능하기 때문에 일반적으로 많이 사용되는 RSS Reader를 이용하였다. 실험을 위한 RSS Reader의 일반적인 구조는 그림 3와 같다.

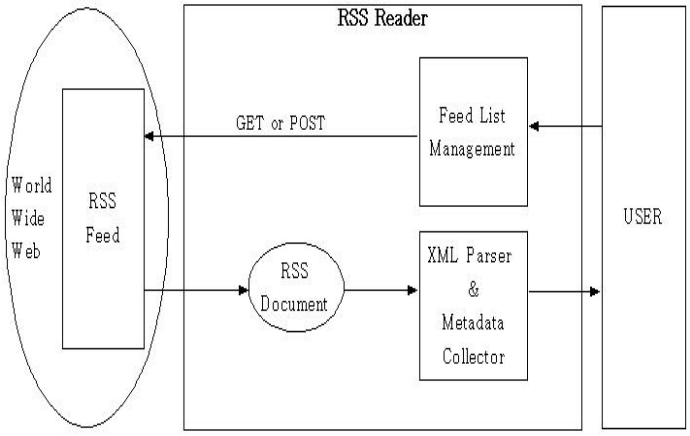

그림 3. RSS Reader 구성도

4.1 중복 확인 실험 설계

중복 되는 메타데이터를 확인하기 위하여 현재 많이 사용되고 있는 RSS Syndication을 이용하였다.
일반적으로 많이 사용되고 있는 RSS Reader를 PC환경에서 설치하여 1일동안 받은 Item들을 정성적 방법으로 비교하였다. Feed주소는 객관성을 위하여 다양한 분야에서 인기있는 사이트 위주로 설정하였다. Reader에 설정된 환경 값은 다음과 같다.

- 수신 주소 개수 : 총 35개의 RSS Feed주소
  (7개 분야별 인기 TOP5순위 사이트)
- 기간 : 6일
- 총 수신 데이터량 (item개수) : 4505개

수신된 Item의 Title을 정렬하고 키워드를 검색한 후, 연결된 사이트 주소를 액세스하고 컨텐츠를 비교하는 방식을 취하였다. 여기서 중복의 의미는 컨텐츠의 100%일치뿐만 아니라 같은 내용을 바탕으로 문단이나 일부분을 수정한 것까지 모두 포함한다. 즉, 같은 내용과 맥락으로 이뤄져 있다면 중복으로 간주하였다.

4.2 확인 실험 결과

그림 3은 사용자 설치 응용프로그램 중 하나인 Fish라는 RSS Reader를 실행시킨 화면이다. Fish 프로그램을 통해 중복 확인을 실험한 결과는 일별 Feed된 내용과 Feed주소의 연관성 유무에 따라 정확한 수치는 조금씩 달라질 수는 있지만, 10%정도의 중복율을 보여줬다. 여기서 중복의 기준과 등록된 Feed에 따라 차이가 발생할 수 있을 수 있지만, 일반적인 RSS 사용자의 패턴을 기준으로 하였으므로 10%에서 크게 벗어나지 않을 것이다.

그림 3. RSS응용 프로그램의 중복 예

4.3 중복 필터링 기능 구현

중복된 메타데이터를 구분하기 위한 방법으로 일반 Text를 자연어 처리한 후, 키워드를 추출하고 이를 비교하는 방식을 택했다. 중복 확인 했던 경우와 마찬가지로 RSS규격문서를 수신하는 방식을 이용하였고, RSS 규격에서 메타데이터에 해당하는 Item을 비교하도록 하였다. 그리고 Item은 일반적으로 Title과 link, Description을 필수요소로 포함한다[5]. 세가지 요소 중 item의 내용을 가장 잘 나타내고 대표할 수 있는 title을 이용하여 실험을 수행하였고, title의 키워드를 추출하여 Item간 매칭 비율을 확인하였다. 중복 확인 신뢰도와 Content 대표성, 그리고 비교 효율성 등을 고려하였을 때, title내 유효 키워드를 이용하는 것이 정성적인 방법으로 가장 효율적이고 정확하다는 결론을 내렸기 때문이다. 또한 정확도를 높이기 위해 보조적인 방법으로 숫자의 일치도를 조사하여 중복 여부를 판단하였다.

유효한 키워드는 조사와 어미를 빼는 자연어 처리 방법을 이용하여 추출하였다. 그러나 한글의 특성상 100% 자연어 처리가 불가능하고, 동음이의어, 유사어가 있어 중복 컨텐츠의 메타데이터에서 키워드가 100% 일치하기는 어렵다. 그래서 중복으로 인정할 수 있는 threshold값을 정할 필요가 있었다.

메타데이터를 이용하는 서비스나 시스템에 따라 threshold값 설정에 차이가 있을 수 있지만, 실험에서는 Feed로부터 받은 RSS Data에서 Title이 하나의 문장으로 되어 있고 한 문장은 5개의 유효 키워드를 가진다고 보고 이 중 동음이의어나 유사어의 1회 발생을 고려하여 Threshold값을 설정하였다. 즉 5개의 유효 키워드 중 4개의 Keyword가 일치하는 80% 일치 이상이면 컨텐츠가 서로 중복되는 것으로 설정 하였다. 또한 Title에 나타나는 숫자는 의미있는 값일 확률이 높다는 가정으로 숫자의 중복여부를 확인한 결과 3개 이상이 겹칠 경우엔 90%이상이 중복 컨텐츠였다. 즉 Title의 키워드 중복 여부 뿐 아니라 추가로 숫자의 일치 여부까지 확인하면 필터링의 정확도가 더욱 높아졌다. 그래서 두 가지 방법이 혼용된 알고리즘을 적용하여 RSS Reader를 구현하였다.

5. 중복 필터링 실험 결과

표 1. 중복 필터링 데이터 측정 결과

	Total Item	발견 Item	미발견 Item		Filtering Item	중복 비율
1회	1331	98	8		1233	0.006488
2회	1332	104	6		1228	0.004886
3회	1343	126	5		1217	0.004108
4회	1234	87	6		1147	0.005231
5회	1320	117	6		1203	0.004988
6회	1320	78	5		1242	0.004026
평균	7880	610	36		7270	0.004955

표 1은 4.1장에 소개된 방식을 통해 중복으로 판명된 Item중 필터링 과정을 거쳐 발견된 Item개수와 미발견된 Item 개수을 표시한 결과값이다. 그리고 필터링 후 남은 item 중 중복된 컨텐츠 비율을 계산하였다.

중복 비율 = 미발견 Item / (Total Item - 발견 Item)

실험 결과 기존의 10.3% 수준의 중복율이 0.5% 수준으로 감소하는 효과를 보였고, 중복 메타데이터를 필터링 하는 능력이 적게는 92% (98개 발견/106개 중복)에서 많게는 96%(126개 발견 /131개 중복)수준까지 나타났다. 미 발견 Metadata의 경우는 자연어 처리가 미흡한 원인으로 발생했다. 예를 들면 Feed목록으로부터 받아온 데이터 내부의 Title에서 같은 뜻을 가진 키워드가 한글과 외래어로 표시되거나 축약어로 나타낸 경우가 있었고 제안된 알고리즘은 의미 구분은 포함하지 않았기에 중복여부를 판단할 수가 없었다. 자연어의 처리상 발생하는 유사어의 미 구별이 원인으로 해결 방법은 자연어 처리시 온톨로지를 이용한 방법 등으로 보완이 가능 하다. [9] [10]

6. 결론

웹 환경에서 엄청나게 많아진 컨텐츠를 효과적으로 제시하기 위한 연구는 향후 지속 될 것이다. 이러한 연구를 위해서 메타데이터를 활용하는 기술이 더욱 발전해야 하고, 발전을 위한 바탕 기술로 XML을 많이 이용하고 있다. 지금까지 본 논문에서는 메타데이터 수신 기술들의 통합을 통해 메타데이터 정보를 폭넓게 이용하고 중복된 내용을 제거하여 다양한 프로그램과 서비스에서 응용이 가능한 효율적인 메타데이터 통합 방법을 제시하였다. 구체적으로 예를 들면 많은 연구가 이루어 지고 있는 개인화 서비스에서 개인의 관심 정보를 키워드로 나타낼 수 있고, 키워드를 바탕으로 사용자가 관심이 있는 컨텐츠를 보기 위해서 컨텐츠에 관한 메타데이터를 받아오게 되는데, 제시된 통합 메타데이터 기술을 활용하여 메타데이터를 수신할 경우 웹 컨텐츠 메타데이터 정보를 풍부하게 받아올 수 있고, 중복 제거를 통해 시스템 효율과 서비스 품질을 높일 수 있다. 앞으로는 온톨로지를 이용한 방법으로[9][10] 키워드의 의미 기반 검색이 가능할 것으로 예상되기 때문에, 온톨로지가 적용되는 경우의 시나리오도 차후 고려해 볼 것이다.

7. 참고 문헌

[1] O. Lassila, ¡Web metadata :a matter of semantics¡, IEEE : Internet Computing, Vol. 2, Issue:4 , pp. 30-37 1998

[2] 권은정, 최동준, 권오형, ¡DVB IPTV 표준화 동향 분석¡, 정보통신 연구 진흥원, 주간 기술동향 통권 1196 호, 2005 년

[3] 정한민, 이미경, 성원경, ¡Open API 기술 동향¡, , 정보통신 연구 진흥원, 주간 기술동향 통권 1296 호, 2007

[4] UserLand, RSS 2.0 Specification, *http://cyber.law.harvard.edu/rss/rss.html* , 2003

[5] Ruey-Shun Chen, Shien-Chiang Yu, ¡Developing an XML framework for metadata system¡, ACM International Conference Proceeding Series, Vol.49, SESSION: Electronic document technology, pp267-272, 2003

[6] Allan Heydon, Marc Najork, ¡Mercator: A scalable, Extensible Web Crawler¡, World Wide Web vol.2, no.4, pp219-229, 1999

[7] Naver OpenAPI website , http://openapi.naver.com

[8] 한국정보 사회 진흥원, ¡참여웹과 사용자 제작 콘텐츠(UCC) : 새로운 가치사슬과 비즈니스 모델¡, 2007

[9] A. Gomez-Perez, O. Corcho, ¡Ontology Languages for the Semantic Web¡, IEEE Intelligent Systems, vol 17, no.1, 2002

[10] Fikes, Richard, Jessica Jenkins, and Qing Zhou, ¡including Domain-Specific Reasoners with Reusable Ontologies¡ Proceedings of the 2004 International Conference on Information and Knowledge Engineering, LasVegas, Nevada, USA, 2003