웹 정보추출의 성능향상을 위한 사용자 관심 부분 추출기의 구현

최철희[0] 홍광희 최중민
한양대학교 컴퓨터공학과
{chchoi, khhong, jmchoi}@cse.hanyang.ac.kr

Implementation of an Extractor of User Selected Parts for Improvement of Web Information Extraction

Cheolhee Choi[0], Kwanghee Hong, Joongmin Choi
Dept. of Computer Science & Engineering, Hanyang University

요 약

인터넷이 발전할수록 정보의 양이 늘어나게 되어 방대한 양의 데이터 속에서 적합한 정보를 추출하는 방법이 필요하다. 그리고 같은 데이터라 하더라도 유용한 정보라고 판단하는 것은 개인의 관심도에 따라 다르다. 따라서 우리는 사용자 관심 정보 추출이라는 목표 아래에서 개인간의 차이에도 영확히 정보를 추출할 수 있는 방법의 필요성을 인지하여 정보추출의 사전 단계에서 사용자가 원하는 정보가 있는 블록을 식별하는 방법에 대해서 연구하였다. 사용자가 선호하는 정보가 들어있는 블록들에 대해서만 정보 추출 기법을 적용하면 정확성과 속도면에서 좋은 결과를 얻을 수 있을 것으로 예상된다. 또한 XML-QL[7]형식의 질의를 통해 사용자의 요구 변화에 유연하게 대처하는 방법을 제안한다.

1. 서 론

인터넷의 발달로 다루는 정보의 양도 증가함에 따라서 사용자가 원하는 정보를 선별적으로 추출할 수 있는 방법이 필요해 졌다. 정보가 존재하는 사이트에 질의를 통해 필요한 정보를 얻는 정보추출기법[1]이 있지만 한 문서당 존재하는 정보의 양이 증가함에 따라서 정보 추출 대상과 관련된 부분을 찾아내는 준비과정으로 페이지 분할(segmentation) 작업이 중요해지게 되었다. 현재의 자동화된 페이지 분할은 사용자에게 편리함을 가져줄 수 있지만 정보의 정확도나 전체 추출 시간은 아직 만족스럽지 못하다. 이 연구의 목적은 정보추출의 사전단계의 과정인 자동화된 페이지 분할 대신 사용자의 개입을 통해 사용자의 의사가 반영된 부분을 명확히 학습에 적용시킴으로써 정보 추출 시스템의 정확도와 성능을 향상시키는 것이다.

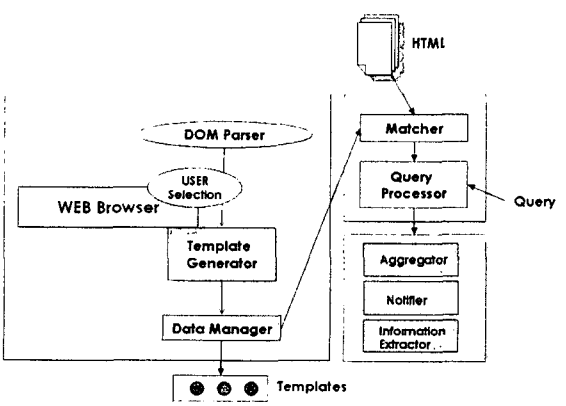

그림 1 시스템 구조

2. 시스템 구조

전체 시스템은 그림1과 같은 구조로 되어 있다. 그림 1에서 왼쪽이 학습 단계라고 생각할 수 있는 템플릿 생성 단계이고 오른쪽 상단이 생성된 템플릿을 적용하는 단계이다. 또한 오른쪽 하단은 이를 활용하는 단계이다.

생성단계에서 시스템은 사용자가 웹 서핑 중 관심 있는 영역을 선택하면 그 영역의 정보를 가지고 시스템은 템플릿을 생성하고 데이터 베이스에 저장한다. 적용단계에서는 생성된 템플릿들과 입력으로 들어온 HTML페이지, 그리고 관련 질의를 가지고 Matcher와 Query Processor가 입력으로 들어오는 페이지와 비교하여 결과를 XML형태로 내보낸다. 마지막으로 활용단계에서는 생성된 XML문서가 정보 추출기나 집계기, 알람기 등의 컴포넌트에서 사용된다.

3. 사용자 관심 섹션

섹션은 그림 2와 같이 사용자가 선택한 태그부터 그 태그와

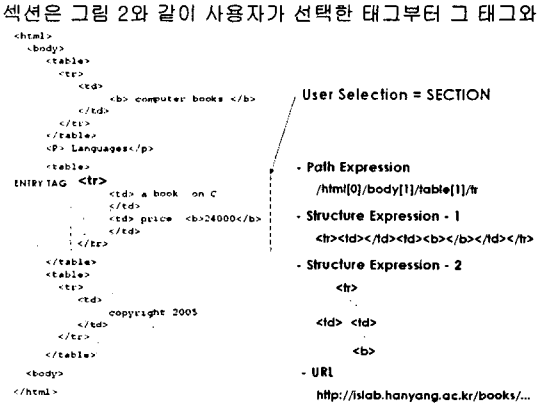

그림 2 템플릿 생성

짝을 이루는 태그까지로, 그 사이에 존재하는 태그들까지 포함하는 가상 블록이다. HTML 페이지는 화면에 보이는 배치를 위해 태그(tag)를 이용한다. 이 시스템에서는 화면의 구분 단위로 사용되는 태그를 사용자의 관심 섹션을 식별하는 단위로 사용한다. 이러한 구별 단위는 자칫 페이지의 분류단위를 너무 확장시켜서 분리하는 의미를 잃을 수도 있지만 현재 대부분의 웹 문서들이 복잡한 태그 정보를 가지고 있다고 가정했고 특히 우리 시스템이 주로 사용되게 될 상업적인 사이트들은 정보가 집중된 부분에서는 정보의 구별단위로 주로 복잡한 태그의 중첩인 표를 사용한다는 것을 발견할 수 있었다.

3.1 사용자의 섹션 선택 방법

이 시스템은 인터넷 익스플로러에 툴바(toolbar)형태로 제공되며 사용자가 웹 서핑 중 선택한 부분을 입력 받아서 템플릿을 생성한다. 이것을 위해 시스템은 페이지를 HTML DOM트리 변환하고 DOM트리 상의 태그들을 순회하면서 선택한 태그와 그 태그의 하위태그들을 블록으로 고려해서 사용자에게 보여주면서 정확한 블록 선택을 돕는다. 이 때 우리는 사용자가 선택한 부분을 태그 단위로 입력 받게 되는데 이러한 태그는 시작과 끝이 있는 중첩 가능한 태그들로 제한된다. 이유는 사용자가 선택하는 부분을 포함하는 최상위 태그를 섹션의 진입점으로 선택하기 위해서 이다.

3.2 템플릿(template)의 생성

템플릿은 사용자의 섹션 태그 선택에 따라서 섹션경로와 섹션 태그 배열 그리고 생성 URL로 이루어 진다. 각각은 입력으로 받은 페이지에서 섹션을 추출할 수 있는 중요 조건이 된다. 각각의 조건은 사용자의 선택에 의해 정해진 가중치를 갖게 되는데 이것은 사용자의 개입으로 보다 정확한 결과를 얻기 위함이다. 물론 실험을 통해서 각각의 가중치를 구할 수 있지만 여기서는 사용자의 참여를 반영하기 위해서 각각의 가중치를 템플릿 생성시에 설정할 수 있다. 가령 태그 배열을 우선할 경우 사용자가 같은 페이지의 반복되는 패턴을 추출 하는 것에 중점을 두는 의도로, 섹션 경로에 우선을 둘 경우 템플릿 생성시와 가장 근접한 위치에 존재하는 섹션을 추출하는 의도로, 추출된 URL에 가중치를 둘 경우 페이지가 많고 분류에 따라 레이아웃이 다양한 사이트에서 사용하려는 의도로 파악될 수 있다.

```
<template>
   <url>http://islab.hanyang.ac.kr/books/...
   </url>
   <path>/html[0]/body[0]/table[1]/tr </path>
   <sequence1>
   <![CDATA[<tr><td></td><td><b></b></td></tr>]]>
   </sequence1>
   <sequence2>
      <row id = 1>tr[0]/td</row>
      <row id = 2>tr[1]/td</row>
      <row id = 3>tr[1]/td/b</row>
   </sequence2>
</template>
```

그림 3 생성된 템플릿 XML파일의 일부분

3.2.1 섹션 경로

웹 페이지의 최상위 태그에서부터 섹션의 진입 태그까지의 경로를 나타낸다. 이때 부모와 자식 사이의 구분은 /로 하고 중괄호에는 해당 자식이 부모의 몇 번째 자식인지에 대한 숫자를 적는다. 단, 이때 0부터 시작하고 자식은 같은 태그 이름을 가진 자식의 숫자를 의미한다. 가령, 섹션경로 table[0]/tr은 table 태그의 tr이라는 이름을 가지는 태그 중에서 첫 번째 자식인 tr을 의미한다. 이렇게 같은 이름을 가지는 태그에 대해서만 생각하는 이유는 페이지 레이아웃의 미미한 변화에 대해서는 유연하게 대처하기 위해서이다.

3.2.2 섹션 태그 배열

섹션내의 태그 집합을 표현하는 방식에는 두 가지 방법이 있다. 첫째는, 섹션내의 태그 집합을 연속적인 순서로 고려하는 방법이다. 이것은 태그 집합을 태그 하나당 하나의 문자로 생각하여 전체 집합을 문자열로 보고 편집거리(edit distance)[5]를 이용하여 태그 집합을 평가하는 방법이다.

두 번째는 섹션내의 태그의 배열순서를 트리로 고려하는 방법이다. 다만 전체 트리의 비교를 하는게 아니라 문자정보가 존재하는 태그 배열에 대해서만, 즉 리프 노드에 내용이 존재하는 태그의 시퀀스들에 대해서만 고려를 한다. 그림3에서 sequence2에 해당하는 부분이 두 번째 방법의 표현법이다. 이것은 텍스트 노드를 가지는 태그까지의 각각의 경로들의 집합이다. 이처럼 분리된 데이터는 뒤에서 다루게 될 질의처리의 조건 비교를 위해 사용되거나, 정보추출의 기법을 적용할 때 정제된 입력으로 제공할 수 있다. 이 시스템에서는 관련도 비교에는 1번의 결과를 문자 집합의 전체 길이로 나눈 값을 사용한다. 그리고 2번의 결과는 질의처리를 위해서 사용되고 있다.

```
DiffPath : input   A,B:path_expr
           output double
BEGIN
  diff = 0.; num = 0; i = 0

  // A[i] : A's ith tagname.
  // A[i].idx : A's ith index
  WHILE A[i] is the ENTRY_TAG
     if (A[i] == B[i]){
        diff += |A[i].idx - b[i].idx|
        i++; num++;
     }else
        return INFINITE (they are different)
  END
  return diff/num;
END
```

그림 4 섹션 경로의 비교 알고리즘

3.2.3 섹션 생성 URL

사용자가 템플릿을 생성했던 URL을 의미한다. 이것은 페이지의 레이아웃이 비슷한 페이지의 경우 그 접두어(prefix)가 같다는 가정하에 평가 요소로 선정되었다. 주로 한 사이트 내의 다른 페이지들 사이에서 같은 레이아웃을 가질 것이라는 가정 때문에 입력페이지의 URL과 템플릿이 생성되었던 URL의 유사도가 높을수록 높은 점수를 갖게 된다. 또한 정보가 밀집된 사이트의 경우 페이지들의 배열을 가지고 있는 주제별로 같은 디렉토리로 묶어서 관리하는데 이러한 URL 정보를 가지고 웹 사용 마이닝(Web Usage Minig)기법 등을 통해 주제별 디렉토리에서 하위 디렉토리의 집계된 정보를 얻을 수 있다. 이것은 카테고리와 유사한 효과를 얻게 되어 온라인 상점이나 뉴스사이트에서 상품이나 뉴스들을 관련 카테고리에 따라서 집계 할 수 있는 효과가 있다. 관련도 비교를 위해 사용될 때는 URL의 질의 문자열(Query String)까지는 구분하지 않고 최종 페이지까지만을 비교 대상으로 삼는다. 그림4의 경로 비교알고리즘과 유사한 방법으로 URL간의 차이 정도를 비교한다.

4. 섹션추출 방법

원하는 정보가 있는 웹 페이지로 사용자가 탐색을 하면 이 시스템의 입력으로 해당 페이지가 들어온다. 그러면 시스템은

해당페이지의 구조를 DOM구조로 메모리에 저장한 후 각 템플릿의 진입점에 해당하는 태그를 찾아서 노드 집합을 만든다. 생성된 노드 집합에서 각 노드에 따른 섹션 경로, 섹션 태그 배열 1,2 그리고 해당 페이지의 URL을 구한다. 이러한 3가지 기준으로 각각을 입력문서와 비교하여 관련도 점수를 구해 결과를 얻거나 추가적인 질의를 통해서 선별된 결과를 얻을 수 있다. 섹션 추출 방법은 단일 섹션 추출과 복합 섹션 추출로 나뉜다.

4.1 단일 섹션 추출

입력으로 들어온 페이지와 설정된 템플릿들과 일치하는 섹션이 있는지 검색하는 과정이다. 앞선 3가지 기준으로 비교하여 관련도 점수가 높게 매칭된 부분부터 낮게 매칭된 부분의 순서로 정렬하여 결과 파일을 생성한다. 한 입력 페이지 안에서 여러 개의 목적이 다른 템플릿을 생성하여 결과를 얻을 때 사용된다.

4.2 복합 섹션 추출

추출을 할 때 논리적으로 관계된 섹션들끼리는 그룹을 지어서 추출해야 할 필요가 있다. 가령 도서 사이트에서 책의 제목과 책의 가격을 각각 다른 템플릿을 생성해서 분리하였을 경우 추출의 결과는 분리되어 있고 추출된 결과들끼리의 연관성은 없다. 경우에 따라서 두 정보가 함께 고려되어야 할 필요가 있으며 이로 인해 각 섹션을 논리적인 단위로 그룹화하여 추출할 수 있는 추출 방법이 필요하다.

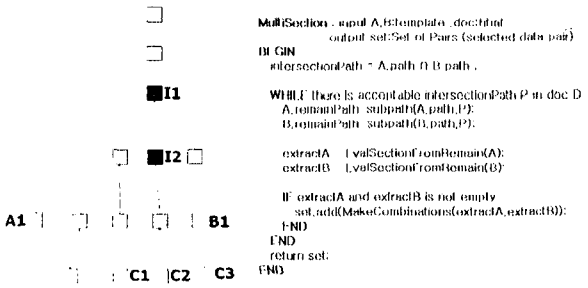

그림 5 복합섹션 추출 알고리즘과 예제

그림 5는 복합 섹션 추출 알고리즘으로 입력으로 템플릿들과 문서를 받아서 해당문서에서 복합섹션으로 추출 가능한 쌍들의 집합을 결과로 준다. 여기서 A, B, C는 각각 템플릿을 의미한다. A1은 A템플릿의 B1은 B템플릿의 C1, C2, C3는 C템플릿의 단일 섹션 추출 부분을 나타낸다. 단일 섹션 추출의 경우 이 5개의 매칭된 인스턴스가 각각의 관련도에 따라 높은 점수대로 결과를 생성할 것이다. 복합 섹션 추출일 경우 사용자가 A와 B템플릿의 연관 관계를 맺으면 알고리즘은 우선 A와 B의 공통 부모인 I1을 찾고 그 아래에서 포함되는 A1과 B1을 찾을 수 있으므로 {{A1,B1}}의 결과를 얻을 수 있다. 같은 방법으로 B와 C가 관계를 맺으면 I2에서 {{B1,C1},{B1,C2},{B1,C3}}의 3개 쌍의 집합을 결과로 준다. 물론 결과 집합이라고 하지만 관련도 정도는 점수로 계산되어 비교할 수 있다. 이러한 복합 섹션은 이 시스템의 사용자 개입이 DOM 트리상의 물리적인 부분에만 관여할 수 있는 한계점을 극복하는 데 도움을 줄 수 있다. 즉, 사용자는 물리적으로는 떨어져 있는 섹션들을 논리적인 그룹으로 선택할 수 있다.

4.3. 조건 질의

이 시스템은 XML-QL[7] 형식의 질의를 통해 추출결과의 생성 조건을 명확히 해서 보다 정확한 결과를 얻을 수 있다. 여기서 조건 문자열인 contains는 부분 문자열의 포함여부를 나타낸다. 이 밖에 산술 비교 연산자도 가능하게 포함시켜서 정밀하게 조건을 주어서 추출 결과를 제한할 수 있다. 아래의 예제는 한국판 *The C Language* 를 포함하는 섹션을 찾는 질의어를 나타낸다.

```
Where
<table>
  <tr>
   <td> $title</>
   <td> $language</>
  </tr>
</table> in URL with template_name,
        $title contains "The C language",
        $language contains "korean"
```

그림 6 XML-QL 형식 질의

5. 결론

웹 페이지 레이아웃은 광고 영역과 내용영역의 구분이 모호하고 내용 영역 안에서도 각각의 사용자마다 각기 다른 성향을 가지고 있기 때문에 관심을 가지는 영역은 개인마다 큰 차이를 보인다. 사전작업으로 사용자의 요구에 근접한 부분으로 그 범위를 제한하면 사용자의 다양한 욕구에 대해서도 적합한 정보를 얻을 수 있는 확률이 증가한다.

단순히 정보추출을 하기 위한 사전 단계로서의 역할 뿐만 아니라 템플릿 정보를 이용하여, 여러 페이지들에 분산되어 있는 관심 부분의 세분화된 집계가 가능해 지며, 이를 통해 기존 정보 추출 시스템이 문서 하나의 단위에서만 추출하게 만들어졌을 경우에도 집계된 자료로 올바른 결과를 얻을 수 있다. 또한 관심 의사를 둔 부분의 변경에 대한 통보 에이전트와 같은 기능의 추가를 통해서 보다 종합적인 시스템으로 발전시킬 수 있다.

참고문헌

[1] A.H.F. Laender et al : A brief survey of web data extraction tools. SIGMOD Rec., 31(2), pp.84-93, 2002.

[2] J. Freire, B. Kumar and D. Lieuwen, WebViews: Accessing Personalized Web Content and Services, WWW10, ACM Press, 2001, pp. 576-586

[3] Zehua Liu, Wee-Keong Ng, Ee-Peng Lim : Personalized Web Views for Multilingual Web Sources, IEEE Internet Computing, vol. 08, no. 4, pp. 16-22, July/August,2004.

[4] Robert Baumgartner, Sergio Flesca, Georg Gottlob: Visual Web Information Extraction with Lixto. VLDB 2001, pp.119-128, 2001

[5] Dan Gusfield, Algorithms on strings, trees, and sequences: computer science and computational biology, Cambridge University Press, New York, NY, 1997

[6] Document Object Model (DOM)
http://www.w3.org/DOM/

[7] XML-QL: A Query Language for XML
http://www.w3.org/TR/1998/NOTE-xml-ql-19980819/