웹의 구조와 웹문서의 적합도를 이용한 효율적인 인터넷 정보추출에 관한 연구

황 인 수[1)]
insoo@jeonju.ac.kr
전주대학교 정보기술학부
전북 전주시 완산구 효자동 3가 1200, 063) 220-2757

키워드 : information retrieval, web crawler, agent

초 록

인터넷을 위한 인프라의 구축이 확대되고 인터넷의 활용이 생활화됨에 따라 웹이나 전자우편을 통해 엄청난 양의 정보가 제공되고 있으며, 그 종류도 뉴스, 광고, 홍보, 커뮤니티 등으로 매우 다양하다. 그러나 각 사용자가 인터넷을 통해 언제든지 접근 가능한 테라바이트(TB) 이상의 엄청난 정보량에 비해 개인이 필요로 하는 정보는 극히 일부분에 지나지 않는 것이 사실이다. 따라서, 인터넷에서 사용자가 원하는 정보를 신속하게 찾아서 제공해주는 인터넷 정보 에이전트(Internet Information Agent)의 역할은 점점 더 증대되고 있다.

인터넷의 정보를 처리하는 에이전트는 크게 정보검색 에이전트, 정보필터링 에이전트, 정보통합 에이전트, 정보추출 에이전트 등으로 분류할 수 있다. 정보검색 에이전트(Information Retrieval Agent)는 인터넷으로부터 사용자가 원하는 정보를 찾아주는 역할을 하는 것으로서, 그 예로는 Web Crawler를 들 수 있다. 정보필터링 에이전트(Information Filtering Agent)는 인터넷을 통해 제공되는 수많은 자료로부터 사용자가 원하는 정보만을 필터링하거나 가공해주는 역할을 하는 것으로서, 그 예로는 전자우편 에이전트 등을 들 수 있다. 정보통합 에이전트(Information Integration Agent)는 여러 가지의 이질적인 정보원으로부터 정보를 검색하여 단일화된 형태로 통합하여 제공하는 역할을 하는 것으로서, 그 예로는 메타 검색엔진이나 비교쇼핑 시스템 등을 들 수 있다.

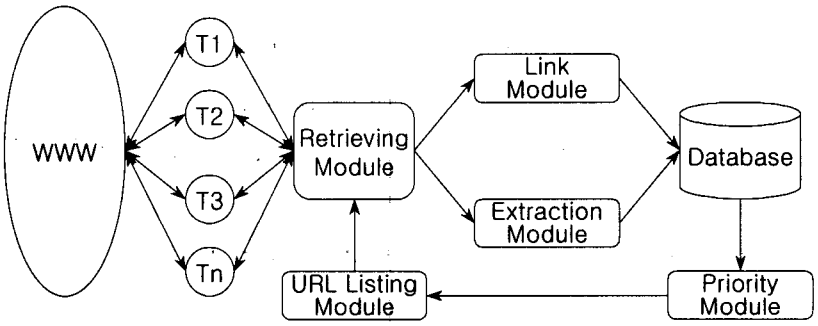
끝으로, 정보추출 에이전트(Information Extraction Agent)는 Wrapper라는 정보추출규칙을 이용하여 인터넷 HTML 문서로부터 사용자가 원하는 부분의 텍스트 정보를 추출하는 역할을 하는 것으로서, 그 예로는 전자우편주소 추출 에이전트를 들 수 있다. 여기서, Wrapper는 정보원으로부터 원하는 정보만을 추출하기 위한 규칙이나 프로그램을 의미하는 것으로, 웹문서마다 그 구성 방식이나 내용이 다르기 때문에 추출하고자 하는 정보의 종류나 웹문서의 구성에 따라 서로 다른 Wrapper를 적용하여야 한다.

본 연구는 위에서 기술한 여러 가지의 인터넷 정보 에이전트 중에서 정보검색을 통한 정보추출 에이전트의 설계 및 구현에 초점을 맞추고 있으며, 정보추출을 위한 Wrapper가 비교적 정형화되어 있는 전자우편주소 추출 문제를 대상으로 하였다. 전자우편주소는 인터넷 웹문서에 텍스트의 형태로 존재하기 때문에 각 문서를 탐색한 후 HTML로 이루어진 텍스트로부터 추출하여야 한다. 그러나, 인터넷에 존재하는 모든 웹문서를 탐색하여 전자우편주소를 추출하는 것은 현실적으로 불가능하므로 효율적인 Web Crawling 전략이

1) 전주대학교 정보기술학부 교수

요구된다.

이에 따라, 본 연구에서는 전자우편주소를 추출하기 위한 Wrapper를 구성한 후, 효율적인 Web Crawling을 위하여 웹의 구조(Hyperlink)와 웹문서(Hypertext)의 적합도를 함께 이용하는 방안을 제안한다. 이의 구조를 그림으로 나타내면 <그림 1>과 같다.



<그림 1> Web Crawling & Information Extraction Agent의 구조

여기서, Extraction Module은 인터넷을 검색하여 수집한 웹문서로부터 전자우편주소를 추출하기 위한 모듈로서, 본 연구에서는 JAVA 언어의 StringTokenizer를 이용하여 다음과 같이 Wrapper를 구현하였다.

```
public void extractMailFrom(String content, Vector mail) {
    String separator="₩n₩r₩t<>: ";
    StringTokenizer st=new StringTokenizer(content, separator);
    while  (st.hasMoreTokens()) {
        String token=st.nextToken();
        if (token.indexOf("@") > 1 &&
            token.indexOf("@")+2 < token.indexOf(".") )
            mail.addElement(token);
    }
}
```

<그림 2> 전자우편주소 추출 Wrapper의 예

위의 <그림 1>에서 Priority Module은 적합도가 높을 것으로 예상되는 웹문서의 링크를 선정하기 위한 모듈이다. 본 연구는 웹문서로부터 전자우편주소를 추출하는 것을 목표로 하기 때문에, 웹문서가 포함하고 있는 전자우편주소의 개수가 많을수록 웹문서의 적합도가 증가하도록 다음과 같이 계산하였다. 여기서, m은 웹문서 p에 존재하는 메일의 개수를 나타낸다.

$$F(p)^* = 1 - \frac{1}{e^m + 1}$$

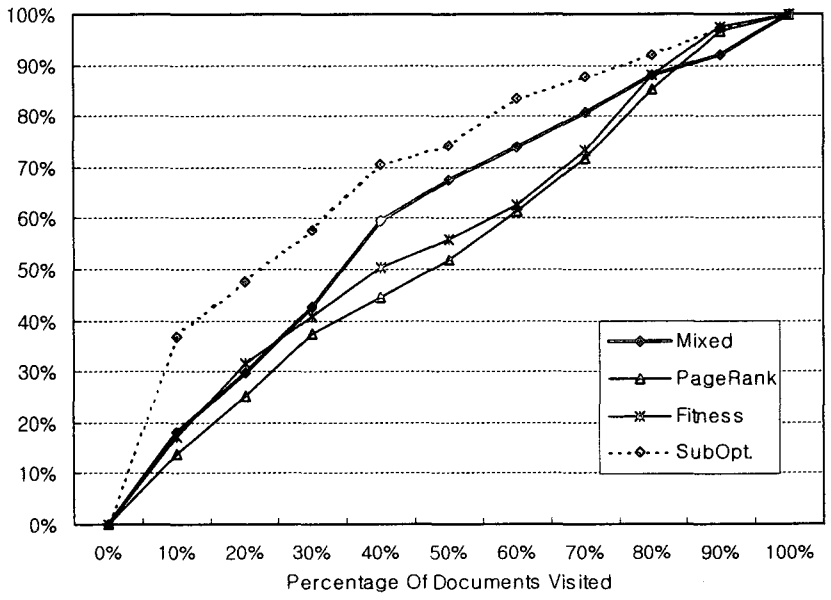
그러나, $F(p)^*$는 웹문서 p를 검색한 후에 사후적으로 결정되기 때문에, 본 연구에서는 PageRank에서 웹문서의 중요도를 다음 링크로 전파하는 기법을 도입하여 이 문서를 참조하는 문서들의 $F(t_i)^+$ 값에 따라 계산한 $F(p)$의 값을 웹문서의 중요도로 사용하였다.

$$F(p) = \sum_{i=1}^{n} \frac{F(t_i)^+}{C(t_i)}$$

여기서, $F(t_i)^+$는 웹문서 t_i를 참조하는 웹문서들에 의해 사전적으로 결정되는 $F(t_i)$와 웹문서 t_i를 방문한 후에 결정되는 $F(t_i)$에 가중치 w를 부과하여 다음과 같이 계산하였다.

$$F(t_i)^+ = w \times F(t_i) + (1-w) \times F(t_i)^*$$

다음은 국내의 P 대학 사이트에 대한 시뮬레이션 결과를 그림으로 나타낸 것으로서 웹의 구조와 웹문서의 적합도를 이용할 경우 Google 등에서 사용하는 PageRank 기법보다 나은 성과를 나타냄을 보여준다. 여기서, SubOptimal은 탐색할 페이지의 적합도를 미리 알고 있다고 가정하여 탐색을 수행한 결과이며, Mixed는 위에서 설명한 Fitness(적합도)에 Location Metric을 가중합하여 탐색을 수행한 결과이다.



<그림 3> 웹구조와 문서의 적합도를 결합한 알고리즘의 성과