인터넷 정보 추출 에이전트

한양대학교 최중민*

1. 서 론

최근 인터넷의 급속한 성장과 보급으로 이메일이나 웹과 같은 정보스트림을 통해 제공되는 정보의 양은 엄청나며, 그 종류도 뉴스정보, 상품 카탈로그, 영화 스케줄 등 매우 다양하다. 하지만 이처럼 웹을 통해 접근 가능한 테라바이트 수준의 정보량에 비해 실제 사용자 개개인이 필요로 하는 정보는 극히 일부분이며, 이러한 현상을 정보과다(information overload)라고 표현할 수 있다. 이 정보과다로 인해 생산성 증대, 교육적 이득, 오락적 가치 등 인터넷을 이용하는 이점이 위협받을 수 있다. 이렇게 인터넷의 정보 홍수 속에서 원하는 정보를 정확하게 제시간에 얻기란 쉬운 일이 아니며, 따라서 이러한 작업을 대신해 주는 에이전트의 역할이 점점 커지고 있다.

본 논문에서는 에이전트의 여러 응용 분야 중에서 인터넷 상의 정보를 처리해주는 정보 에이전트를 기술하고자 한다. 인터넷상에서의 정보를 처리하는 에이전트를 크게 정보검색 에이전트, 정보필터링 에이전트, 정보통합 에이전트, 정보추출 에이전트의 네 가지로 분류할 수 있다 정보검색 에이전트(information retrieval agent)는 사용자가 원하는 정보를 찾아주는 역할을 수행하며 검색엔진(검색로봇)이 대표적인 예가 된다. 정보필터링 에이전트(information filtering agent)는 사용자의 구미에 맞도록 정보를 가공하고 걸러준다. 정보통합 에이전트(information integration agent)는 이형질의 여러 정보소스로부터 정보를 검색하여 단일화된 형태로 통합하여 보여주는 작업을 수행한다. 메타 검색엔진이나 비교 쇼핑 시스템들이 대표적인 예가 된다. 정보추출 에이전트(information extraction agent)는 인터넷 문서에서 원하는 부분 텍스트 정보를 추출해 내는 작업을 수행하며 wrapper라 불리는 추출 규칙을 각 정보소스에 대하여 생성하여야 한다. 정보추출 에이전트는 정보통합 에이전트와 밀접한 관계가 있다. 본 논문에서는 이 네 가지 정보 에이전트를 기술하고 특히 정보추출 에이전트에 초점을 맞추고자 한다.

본 논문은 다음과 같이 구성된다. 1장에서는 정보 에이전트의 필요성과 네 가지 종류의 정보 에이전트를 소개하였다. 2장은 정보검색 에이전트를, 3장은 정보필터링 에이전트를, 4장은 정보통합 에이전트를 각각 기술한다 5장은 정보추출 에이전트를 상세히 기술한다. 6장에서는 결론과 앞으로의 방향을 제시한다.

2. 정보검색 에이전트

정보검색 에이전트는 사용자가 원하는 정보를 찾아주는 역할을 수행하며 검색엔진이 대표적인 예가 된다. 검색엔진은 검색로봇(search robot), 인덱스(index), 질의서버(query server)의 요소로 구성된다. 검색로봇은 주기적으로 웹 공간에 존재하는 문서를 수집하여 인덱싱할 수 있도록 도와준다. 인덱스는 검색로봇이 모아준 문서를 데이터베이스에 저장하는 작업을 한다. 질의서버는 사용자의 질의 검색어를 입력으로 받아서 인덱스를 참조하여 검색결과를 출력해준다.

* 종신회원

검색엔진을 구성할 때 다음과 같이 몇 가지 고려해야 할 이슈가 있다. 첫째는 검색로봇의 항해 전략이다. 인터넷에 존재하는 문서는 대부분 하이퍼링크를 이용하여 다른 정보사이트와 연결되어 있는데 인덱싱을 위해서는 하나의 문서에서 출발하여 그 문서 내에 있는 여러 링크를 어떠한 순서로 검색할지 결정하여야 한다. 대표적으로 깊이우선, 너비우선, 최적우선 등의 방법을 사용한다. 최적우선 방법은 휴리스틱을 이용하여 다음 인덱스할 링크를 결정하는 것으로 많이 쓰이는 휴리스틱 중의 하나는 링크 URL의 길이를 비교하여 작은 길이의 URL을 가진 링크를 우선으로 검색하는 방법이다. 그 이유는 URL의 길이가 작을수록 한 호스트의 최상위 레벨의 위치를 나타낼 가능성이 많으므로 좀 더 광범위한 인덱싱이 될 수 있기 때문이다. 두 번째 이슈는 검색로봇 문제로써 근본적으로 로봇 자체가 링크 URL의 특성을 인식하지 못하는 데서 발생한다. 즉, CGI에 관계된 URL이나 임시로 만들어 놓은 링크에 대한 URL은 인덱싱을 할 필요가 없지만 로봇이 이를 판단하기는 어렵다. 이에 따른 부수적인 결과로는 같은 CGI URL을 무한히 반복적으로 접근하기도 하고 한 호스트의 URL만을 계속적으로 접근하여 그 호스트의 기능을 마비시키기도 한다. 이를 해결하기 위해서 로봇제외 표준(robot exclusion standard)이 제안되기도 하였다. 이것은 각 웹 호스트에 robots.txt라는 파일을 저장하여 로봇이 인덱스할 필요가 없는 문서에 대한 정보를 미리 제공하는 것이다. 즉, 검색로봇은 각 호스트로부터 robots.txt라는 파일이 있는지를 먼저 확인하고 그 파일을 참조하여 CGI나 임시 문서에 대한 인덱싱은 피할 수 있는 것이다. 하지만 이 방법의 문제점은 검색 로봇의 개발자가 아닌 각 웹 정보 사이트의 관리자가 robots.txt 파일을 만들고 수시로 점사해야 하기 때문에 이 표준을 따르지 않는 정보 사이트에는 적용되지 못한다는 것이다. 실제로 통계적으로 이 표준을 따르는 곳은 얼마 안된다는 조사도 있다.

3. 정보필터링 에이전트

정보필터링 에이전트는 사용자의 구미에 맞도록 정보를 가공하고 걸러준다. 정보필터링은 기본적으로 끊임없이 유입되는 정보 중에서 필요한 것이 무엇이고 필요없는 것이 무엇인지를 판단하여 필요하지 않은 것은 무시하는 개념이다. 정보필터링에서는 사용자가 관심을 가지는 사항에 대한 정보를 가지고 있는 사용자 프로파일이 중요한 역할을 한다. 정보필터링 과정은 이메일이나 뉴스그룹의 정보와 같은 정보스트림을 사용자의 프로파일과 비교하여 관심이 있는 정보만을 걸러서 저장한 후 사용자가 볼 수 있게 한다. 사용자는 정보필터링 과정을 거친 결과를 본 후 그것이 실제로 자신이 원하는 것이었는지를 알려주게 되는데 이를 관련성 피드백이라 하며 이 과정을 거치면서 사용자 프로파일을 재구성한다

정보필터링에 수반하는 문제점은 다음과 같다. 첫째, 단어선택 문제인데 정보검색이나 정보필터링 모두에 해당하는 문제이다. 대부분 관심도를 단어로 표현한다고 할 때 같은 관심분야라고 하더라도 사람마다 선택하는 단어가 다를 수 있고, 심지어는 같은 사람이라도 시간 경과에 따라 다른 형태로 표현할 수 있는 것이다. 둘째, 문서 구조화가 되어있지 않거나 일부만 되어있다는 문제이다. 가령 전자우편의 경우 보낸 사람이나 제목 등의 경우는 구조화되어 있다고 볼 수 있지만 내용의 경우는 일반 텍스트의 나열이기 때문에 내용을 통해 필터링 작업을 하는 것은 매우 어려운 일이다. 또 사용자에게 유입되는 정보의 종류가 다양하고 각각이 서로 다른 구조를 가지고 있기 때문에 이를 모두 고려하는 작업이 중요한 이슈로 대두된다. 셋째, 정보필터링 에이전트를 훈련시켜야 한다는 것이다. 사용자의 프로파일은 처음부터 사용자의 의도를 완벽하게 나타낼 수 없기 때문에 점진적으로 만족스러운 상태로 재구성해야 하는데 이를 위해 사용자의 피드백이나 사용 습성에 따라 정보필터링 에이전트를 훈련시켜야 한다. 이 과정은 오랜 시간이 걸리기 마련이며 또한 중간에 사용자의 의지에 의해 관심도가 급격히 바뀌었을 경우는 다시 오랜 기간에 걸쳐 훈련이 되어야 한다.

다수의 정보필터링 에이전트 시스템이 연구용 또는 상용으로 제시되었다. 정보필터링 에이전트는 어떤 정보를 대상으로 필터링 작업을 하는가에 따라 분류될 수 있는데 크게 웹문서 필터링 에이전트, 상용뉴스 필터링 에이전트, 유즈넷뉴스 필터링 에이전트로 나눌 수 있다. 웹문서 필터링

에이전트의 예로는 WebFilter[1], Webcatcher[2], Point Subscription[3], Smart Marks[4] 등이 있고, 상용뉴스 필터링 에이전트에는 NewsHound[5], Farcast[6], PointCast Network(PCN)[7] 등이 있으며, 유즈넷뉴스 필터링 에이전트에는 NewsClip[8]과 SIFT[9] 등이 있다.

4. 정보통합 에이전트

정보통합 에이전트는 인터넷에서 제공되는 다수의 정보 사이트에서 사용자가 원하는 정보를 추출하여 하나의 형태로 제공하는 기능을 수행한다. 정보통합 에이전트의 필요성은 여러 가지로 기술할 수 있지만 그 중에서도 다수의 정보소스를 사용자가 하나 하나 접근하여 검사하는 노력을 줄여주고 각 정보 사이트에서 사용자에게 불필요하다고 판단되는 것을 걸러주는 점을 들 수 있다. 따라서 앞장에서 기술한 정보검색 에이전트와 정보필터링 에이전트의 개념을 가지고 있으며 메타검색 엔진과 같은 개념도 정보통합 에이전트와 맥락을 같이 한다고 볼 수 있다. 정보통합 에이전트는 각 정보소스에 대한 정보추출 규칙을 가지고 있어서 사용자의 질의가 각 정보 사이트의 입력에 맞는 형태로 변환되고 각 사이트에서 처리한 결과를 통합한 후 사용자에게 필요한 정보만 보여준다. 사용자가 출력된 정보를 바탕으로 더 자세한 사항을 파악하기 위해서 해당 정보사이트로 다시 접근할 수 있는 기능도 지원한다.

정보 통합 에이전트의 또 다른 예로는 비교쇼핑 시스템을 들 수 있다. 최초의 비교쇼핑 시스템인 BargainFinder[10]는 온라인으로 음악 CD 정보를 파악하여 주문할 수 있는 여러 개의 CD 정보사이트를 연결하여 사용자의 구매편의를 도모한다. 사용자는 앨범의 제목이나 가수 이름을 입력하게 되는데 BargainFinder에서 이를 각 정보 사이트의 입력 형태로 변환한 후 각 정보사이트에서 돌아온 결과 중에서 CD의 판매가가 나온 부분만 모아서 사용자에게 보여주고 사용자는 이 결과를 보고 한 곳을 결정한 후 더 자세한 사항을 보기 위해서는 해당 정보를 클릭함으로써 개별적인 사이트로 접근할 수 있다.

5. 정보추출 에이전트

정보추출은 한 문서에서 그 문서의 중심적 의미를 나타내는 특정 구성요소를 인식하여 추출하는 작업을 가리킨다[11]. 정보추출의 예로는 날씨 정보를 제공하는 웹 문서로부터 지역, 날짜, 최고온도, 최저온도, 습도 등의 정보를 뽑아내거나, 또는 아파트 정보 문서로부터 방의 개수, 매매가, 전세가, 전화번호 등을 추출하는 것을 들 수 있다.

정보추출의 성능은 확장성(scalability)의 정도에 달려있다. 정보추출 시스템은 대부분 특정 문서에 대해서 의미정보를 뽑아낼 수 있는 추출규칙(extraction rule)을 이용하게 된다. 서로 다른 문서에 대해서 이러한 추출규칙이 어떠한 역할을 하는가에 따라 확장성의 정도가 결정된다. 만일 특정 문서마다 일일이 새로운 규칙을 만들어야 하는 경우 같은 규칙을 다른 문서에는 적용할 수 없기 때문에 확장성이 떨어진다. 수동적으로 규칙을 구성하는 대부분의 시스템이 이 부류에 속한다. 확장성을 가지기 위해서는 일반적인 프로시저 또는 프로그램이 존재해서 처음 접하는 문서에 대해서도 이 프로그램을 통해 자동적으로 추출규칙을 얻어낼 수 있어야 한다. 확장성을 위한 프로그램은 대부분 기계학습(machine learning) 기법을 이용한다.

이 장에서는 정보추출 시스템을 기술하고자 한다. 정보추출 작업이 무엇인지 인식하고, 기계학습을 통한 확장성을 가진 정보추출 기법을 소개하며, 개발된 정보추출 시스템들을 비교하고자 한다.

5.1 인터넷 정보추출

정보추출은 인터넷을 다루는 여러 가지 전략을 가능하게 만드는 기술로서, 이런 전략의 한 예로는 이형질(heterogeneous)의 인터넷 정보소스에 대해 단일화된 뷰(unified view)를 보여주는 정보통합 시스템을 들 수 있다. 예를 들면, 영화에 대한 정보를 제공해주는 통합 시스템에서는 인터넷에 여러 사이트에서 제공하는 영화평이나 출연자, 상영스케줄 등에 대한 서로 다른 형태의 영화에 대한 정보들에 대한 단일한 질의 인터페이스(query interface)를 제공하게 된다. 사용자는 이 인터페이스에 질의를 하게 되고, 통합 시스템은 이 질의를 각 정보 소스에 맞도록 변화시켜 보낸 후 얻어진 결과를 통합하여 사용자에게 보여준다.

정보통합 시스템은 인터넷 사이트를 일반 텍스트로 해석하지 않고 데이터베이스와 유사한 구조적 지식소스(structured knowledge source)로 간주한다. 사이트의 문서로부터 관련 텍스트 부분을 추출하기 위해 정보통합 시스템은 HTML 태그나 광고와 같은 부수적인 정보는 제거하고 내용을 이루는 텍스트만을 대상으로 동작한다. 정보통합 시스템은 wrapper라고 부르는 추출규칙을 사용하는데, wrapper는 한 정보소스마다 하나씩 존재하며 그 정보소스로부터 원하는 정보를 추출하는 규칙이나 프로그램으로 구성되어 있다.

인터넷의 정보추출 작업은 본질적으로 확장성이 없는데, 그 이유는 다음 세 가지로 요약할 수 있다. 첫째, 정보소스의 문서들은 원칙적으로 사람들이 읽기 편하도록 작성되었기 때문에 프로그램이 쉽게 처리할 수 있도록 문서 구성시의 포맷 관행 등에 대한 정보를 제공해주는 사이트는 거의 없다. 둘째, 한 사이트에서 사용된 독특한 포맷 관행이 다른 사이트에도 적용될 가능성이 거의 없기 때문에 사이트가 추가되는 경우 새로운 wrapper가 구성되어야 한다. 셋째, 사이트들이 자주 포맷을 바꾸기 때문에 이전에 만들었던 wrapper가 동작하지 않게 된다. 이러한 어려움에도 불구하고 최근의 연구 결과들을 보면 확장 가능한 정보추출 시스템을 구성할 수 있다는 가능성을 보여주고 있다. 이를 달성하기 위한 열쇠가 되는 기법은 바로 기계학습을 이용하여 자동적으로 사이트에 독특한 추출 규칙을 생성하는 것이다.

여기서 기술하는 인터넷 정보추출 기법의 대상이 되는 것은 주로 HTML로 작성된 문서이다. <h1>이나 <p>와 같은 대부분의 HTML 태그는 의미가 있기보다는 단순히 화면에 출력되는 포맷을 지정하기 위한 것이기 때문에 정보추출에 별로 도움이 되지 못한다. 따라서 HTML 문서로부터의 정보추출은 상당히 어려운 작업임에 틀림없다. 만일 현재 부상하고 있는 XML로 문서가 작성된다면 인터넷 정보추출 작업은 한결 쉬워질 수 있다. 왜냐하면 XML의 태그는 <name>이나 <booktitle>등과 같이 의미를 가지고 있기 때문에 태그만 인식하여도 어느 정도 필요한 내용을 추출할 수 있기 때문이다. 하지만 아직은 XML을 표준으로 간주하는 사이트가 아직은 별로 많지 않기 때문에 HTML 문서에 대한 현재의 정보추출 기법은 당분간은 매우 중요한 역할을 할 것이다. 또한 XML에 대한 정보통합 시스템은 XML 태그의 의미를 인식하기 위해 도메인 지식을 포함하는 온톨로지(ontology)를 가지고 있어야 한다는 제약사항이 존재한다.

우리가 관심을 가지고 있는 정보추출 작업은 인터넷 페이지에 내포되어 있는 데이터베이스와 유사한 구조를 찾는 것이며, 이것은 주로 자연어 처리 분야에 뿌리를 두고 언어적 자질(품사나 어휘적 정보 등)에 의존해온 전통적인 정보추출 시나리오와는 여러 면에서 다르다. 인터넷 정보추출은 단순한 통사적 정보(syntactic information)만 가지고도 원하는 텍스트를 인식할 수 있는 경우가 많다. 특히 대부분의 추출 항목은 시각적으로 구별되는 특성을 가지고 있다. 예를 들면, 영화제목을 항상 빨간색으로 출력하는 사이트의 경우 HTML 문서에서 <FONT color=red>와 </FONT> 사이에 있는 텍스트가 바로 영화제목이 될 것이다.

당연히 이러한 규칙성에만 의존하는 wrapper는 일반적이라고 할 수 없다. 그럼에도 불구하고, 이러한 접근 방법은 상당히 많은 수의 인터넷 사이트에 대해 적용가능하며 실제로 Jango[12]나 Junglee[13]와 같은 상용 인터넷 통합서비스에서 사용되고 있다. 하지만 이러한 통사적 규칙만 가지고는 인터넷 정보추출이 어려울 수 있으며, 확장성이 더욱 중요하게 대두된다. 인터넷 사이트가 선택할 수 있는 포맷 스타일의 수는 매우 많으며 그 스타일 중에서 임의로 하나를 선택했기 때문에 그 선택 자체는 추출 작업에 별 도움을 주지 못한다. 예를 들어, 어떤 사이트는 영화제목을 빨간색의 볼드체로 출력할 수 있고, 다른 사이트에서는 영화제목을 녹색의 이탤릭체로 출력할 수 있다. 또한 같은 스타일이라고 하더라도 여러 가지 다른 방법으로 기술할 수도 있다. 예를 들어, <FONT color=red><BIG>은 <FONT size=+1 color= #ff0000>와 똑같은 효과를 얻는다.

5.2 인터넷 정보 추출을 위한 학습

이렇게 아주 다양하고 임의적인 스타일 선택이 있기 때문에 일반적으로 모든 사이트를 커버할

수 있는 확장성을 가진 정보통합 또는 정보추출 시스템을 구성하기 위해서는 필수적으로 wrapper induction이라고 불리는 기계학습 기법을 이용해야 한다. wrapper induction 단계는 오프라인으로 수행되며, 추출을 원하는 텍스트 부분(text fragment)이 개발자에 의해서 표시된 몇 개의 예제 페이지를 바탕으로 wrapper를 학습하여 구축하게 된다. 학습된 wrapper는 온라인으로 실제 페이지의 내용을 추출하는데 이용된다.

모든 기계학습 응용에서와 마찬가지로 학습되는 표현방법의 복잡도에 따라 학습이 가능한지가 결정된다. 이 문맥에서 표현방법의 복잡도는 wrapper의 정교함과 관계가 있다. wrapper를 연구하는 사람들은 단순한 wrapper로부터 아주 복잡한 wrapper까지 다양한 범위의 wrapper에 대한 학습 알고리즘을 개발하였다. 가장 단순한 wrapper는 문서 페이지가 각 속성의 처음과 끝을 가리키는 신뢰할 수 있는 구별자(delimiter)가 있다고 가정한다. 이 환경에서의 wrapper induction은 추출하고자 하는 텍스트 부분의 바로 앞에 나타나는 텍스트의 공통 접두부(common prefix) 또는 바로 다음에 나타나는 텍스트의 공통 접미부(common suffix)를 찾아내는 것이다. 이런 단순한 접근방법 만으로도 많은 실제 사이트에 대해 성공적으로 wrapper가 만들어졌다.

단순한 wrapper로 처리가 되지 않는 사이트는 좀 더 복잡한 형태의 접근방법이 필요하다. 예를 들어 영화제목을 표시할 때 큰 사이즈의 빨간색 폰트나 녹색 폰트를 사용하지만, 큰 사이즈의 빨간색 폰트를 영화제목과 관련없는 다른 항목을 표시할 때도 사용하는 사이트가 있다면 단순한 규칙만 가지고는 영화제목의 내용을 추출하기 어렵다. 복잡한 wrapper는 이러한 어려움을 다음과 같은 방법으로 해결한다. 첫째, 추가적인 구별자를 도입하여 페이지의 내용과 관계되는 부분과 그렇지 않은 부분(예를 들어 <TABLE> 태그의 앞에 나타나는 텍스트)을 구별한다. 둘째, 논리합(disjunction)을 허용한다. 예를 들어 영화제목의 추출 규칙이 "큰 사이즈의 빨간색 폰트 혹은 큰 사이즈의 녹색 폰트"에서와 같이 "혹은"을 포함할 수 있다. 셋째, 구별자의 개념을 더 일반화하여 구별자만을 검사하는 것이 아니라 추출될 텍스트 자체도 검사하는 것이다. 예를 들어, 큰 사이즈의 빨간색 폰트가 영화제목이 되려면 제목이 대문자여야 한다는 추가 제한을 둘 수 있다. 넷째, 자동으로 페이지를 일차식 표현(first-order representation)으로 바꾸고 귀납적 논리 프로그래밍을 이용하여 임의의 일차식 이론을 학습한다.

Wrapper 생성과 인터넷 문서의 정보추출에 대한 대표적인 연구로는 ShopBot[14], HLRT[15], ARIANDNE[16], WHIRL[17] 등이 있다. 이 중 ShopBot은 wrapper induction을 비교쇼핑 도메인에 적용하여 자동으로 쇼핑몰 사이트의 상품정보 추출을 위한 wrapper를 학습한다. 하지만 규칙성이나 바이어스를 강하게 사용하기 때문에 다룰 수 있는 쇼핑몰이 제한된다. Kushmerick은 여러 wrapper class를 제안하였고 HLRT는 이 class 중의 하나이다. 이 기법은 웹 문서에 국한하지 않고 여러 도메인의 문서에서의 정보추출을 가능하게 하고 있지만 wrapper class 들이 단순하기 때문에 속성이 없는 노이즈 포함문서를 다룰 수 없다.

6. 결론 및 전망

본 논문에서는 인터넷 정보를 가공하고 처리해 주는 인터넷 정보 에이전트를 정보검색, 정보필터링, 정보통합, 정보추출의 네 가지 에이전트로 분류하여 각각의 개념과 응용 시스템 등을 살펴보았고, 특히 정보추출 시스템에 많은 초점을 맞추었다.

정보추출에 대한 많은 연구 결과가 나와있지만 아직까지 임의의 인터넷 사이트에 대해 처리가 가능한 범용의 정보추출 시스템의 개발은 요원하다. 앞으로 다음과 같은 진보가 이루어져야 할 것으로 생각된다. 첫째, 인터넷 정보추출을 위해 자연어 처리 그룹에서 개발한 정보추출 기법을 도입해야 할 것이다. 이 기법을 이용하려면 인터넷 사이트가 언어적 구조를 나타내주어야 한다는 제한조건이 있기는 하지만 이러한 자연어 처리 기법은 데이터베이스와 유사하지 않은 일반 텍스트에 대한 wrapper를 구축하기 위해서는 아주 중요한 것이다. 바람직한 방법은 언어적 자질과 통사적 자질을 혼합하는 접근 방법을 채택하는 것이다. 둘째, 상대적으로 단순한 wrapper는 잘 이해되지만 좀 더 표현력이 있는 wrapper의 복

잡도를 이해하려면 추가적인 작업을 해야 한다. 주어진 도메인을 처리하기에 충분한 가장 단순한 기법을 예상하는 이론을 개발하는 것이 이상적이다. 이러한 노력을 도와주기 위해 RISE 사이트[18]에서는 표준 데이터를 제공하여 이것을 이용해서 정보추출 시스템을 평가하도록 하였다. 셋째, 대부분의 기법이 결국은 자동 프로그래밍(automatic programming)이 될 것이고, 이것의 출력이 바로 정보추출 작업을 수행하는 wrapper가 된다. 하지만 기계학습은 다른 가능성들을 제공하는데 그 중의 하나가 nearest-neighbor 알고리즘으로 추출될 요소의 후보들이 자질 공간(feature space)에 분포시켜 추출할 것과 추출하지 않을 것을 분리하는 방법이다. 이 접근방법은 오프라인으로 훈련시키는 단계가 없으며 정보추출 시스템의 성능을 점진적으로 향상시킨다. 넷째, wrapper induction 작업은 상대적으로 좁은 의미로 해석되어 왔다. 예를 들어 포맷이 바뀔 경우 현재 동작하는 wrapper를 무효화시킬 수 있기 때문에 wrapper induction은 복잡해질 수 있다. 아마도 wrapper 유지보수(wrapper maintenance)가 wrapper를 처음부터 다시 학습시키는 것보다는 쉬울 것이다. wrapper induction은 문서내 추출(within-document extraction)에 초점을 맞추어왔는데 앞으로 문서와 문서 사이의 관계도 중요한 역할을 할 것이다.

참고문헌

[1] WebFilter, http://ils.unc.edu/webfilter.

[2] Webcatcher, http://plum.tuc.noao.edu/webcatcher/webcatcher.html.

[3] Point Subscription, http://www.pointcom.com.

[4] Smark Marks, http://www.netscape.com/comprod/smartmarks.html.

[5] NewsHound, http://www.sjmercury.com/hound.htm.

[6] Farcast, http://www.farcast.com.

[7] PointCast Network, http://www.pointcast.com.

[8] NewsClip, http://www.clarinet.com/newsclip.html.

[9] SIFT(Stanford Information Filtering Tool), http://sift.stanford.edu/.

[10] BargainFinder, http://bf.cstar.ac.com/bf.

[11] N. Kushmerick, "Gleaning the Web," IEEE Intelligent Systems, vol. 14, no. 2, pp. 20-22, 1999.

[12] Jango, http://jango.excite.com.

[13] Junglee, http://www.junglee.com.

[14] R. Doorenbos, O. Etzioni, D. Weld, D, "A Scalable Comparison-Shopping Agent for the World Wide Web," First International Conference on Autonomous Agents, pp. 39-48, 1997.

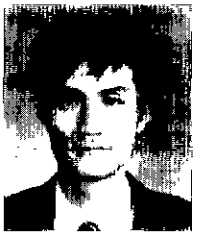
[15] N. Kushmerick, D. Weld, R. Doorenbos, "Wrapper Induction for Information Extraction," International Joint Conference on Artificial Intelligent, pp. 729-735, 1997.

[16] J. Ambite, N. Ashish, G. Barish, C Knoblock, S. Minton, P. Modi, I. Muslea, A. Philpot, S. Tejada, "ARIADNE: A System for Constructing Mediators for Internet Sources," ACM SIGMOD International Conference on Management of Data, pp. 561-563, 1998.

[17] W. Cohen, "A Web-based Information System that Reasons with Structured Collections of Text," Second International Conference on Autonomous Agents, pp. 400-407, 1998.

[18] RISE Repository, http://www.isi.edu/~muslea/RISE.

최 중 민



1984 서울대학교 컴퓨터공학과(학사)
1986 서울대학교 대학원 컴퓨터공학과(석사)
1993 State Univeristy of New York at Buffalo, Computer Science(박사)
1993 ~ 1995 한국전자통신연구소 인공지능 연구실 선임연구원
1995 ~ 현재 한양대학교 전자계산학과 조교수

관심분야 지능형 에이전트 시스템, 인공지능, 정보검색, 데이터베이스, HCI
E-mail jmchoi@csc.hanyang.ac.kr