구조화된 웹 문서에 대한 자동 정보추출

Automatic Information Extraction for Structured Web Documents

윤 보 현*
Bo Hyun Yun

요 약

본 논문에서는 구조화된 웹문서에서 자동으로 정보를 추출하고 추출된 정보를 통합하는 정보추출 시스템을 제안한다. 제안한 시스템은 레이블(label)이 없는 엔티티를 인식하기 위해 확률 기반 엔티티 인식 방법을 이용하며, 추출된 데이터를 이용하여 기존의 도메인 지식을 반자동으로 확장하는 기능을 제공한다. 게다가 기본 페이지에 링크된 하위 링크의 정보를 추출하는 기능을 제공하며, 도메인에 대한 이종의 정보 소스로부터 얻어진 유사 추출 결과를 통합하는 기능을 제공한다.

실험 결과, 도메인 지식만을 이용하여 웹 정보추출 시스템을 평가하였을 경우의 성능에 비해 하위링크의 정보를 추출하거나 확률 기반으로 레이블을 추론하여 추출 시스템을 평가한 경우의 성능이 상당히 향상됨을 보인다. 아울러 본 논문에서 제안하는 웹 정보추출 시스템은 도메인별로 시스템을 융통성 있게 적용시킬 수 있기 때문에 보다 다양한 정보들을 추출할 수 있다. 자동 도메인 지식의 확장이나 확률적 엔티티 인식 방법은 도메인 지식을 이용하는 프로그램이 추출할 수 있는 정보의 질을 증대시키기 때문에, 사용자의 만족도를 극대화시킬 수 있다는 장점이 있다. 따라서 본 시스템은 인터넷상의 영화 사이트나 공연 사이트, 혹은 음식점 사이트에 대해서 정보를 추출해서 사용자의 지적 호기심을 충족시켜줄 수 있을 뿐만 아니라, 다양한 비교 시스템을 구축할 수 있기 때문에 전자 상거래의 활성화에도 기여한다.

Abstract

This paper proposes the web information extraction system that extracts the pre-defined information automatically from web documents (i.e. HTML documents) and integrates the extracted information. The system recognizes entities without lables by the probabilistic based entity recognition method and extends the existing domain knowledge semiautomatically by using the extracted data. Moreover, the system extracts the sub-linked information linked to the basic page and integrates the similar results extracted from heterogeneous sources.

The experimental result shows that the system extracts the sub-linked information and uses the probabilistic based entity recognition enhances the precision significantly against the system using only the domain knowledge. Moreover, the presented system can the more various information precisely due to applying the system with flexibleness according to domains. Because both the semiautomatic domain knowledge expansion and the probabilistic based entity recognition improve the quality of the information, the system can increase the degree of user satisfaction at its maximum. Thus, this system can satisfy the intellectual curiosity of users from movie sites, performance sites, and dining room sites. We can construct various comparison shopping mall and contribute the revitalization of e-business.

☞ Keyword : 정보추출, 엔티티인식, 랩퍼

1. 서 론

네트워과 컴퓨팅 기술의 발전으로 인해 인터넷은 급속도로 성장하게 되었고, 그 결과 이용할 수 있는 정보의 양이 급증하게 되었다. 숙련된 컴퓨터 기술의 필요 없이도 누구나가 인터넷에 홈페이지라는 것을 가질 수 있게 되었고, 다양한 정보 서비스 업체들은 자신들의 고유 영역을 넓혀 가면서 다양한 컨텐츠 서비스들을 제시하고 있는 상황이다. 그러나 이와 같이 정보가 급증하면서 사용자가 얻을 수 있는 정보의 양이 많아진 것은 사실이지만 실제적으로 원하는 핵심 데이터들을 찾기는 좀 더 어려워진 상황이다. 이러한 정보 과부하는 사용자

* 정 회 원 : 목원대학교 컴퓨터교육과 교수
ybh@mokwon.ac.kr(제 1저자)
[2004년/02/11 투고 - 2004/02/25 심사 - 2004/09/20 심사완료]

로 하여금 인터넷 이용에 대한 만족도를 떨어뜨리고 있다. 즉, 인터넷이 아직은 양적인 성장에 비해 질적인 성장이 더딘 상황이라고 볼 수 있다. 이러한 상황에서 사용자가 원하는 정보만을 추출하여 제시하는 시스템에 대한 필요성이 대두되고 있다.

정보 추출의 목적은 많은 내용을 포함하고 있는 문서에서 사용자가 관심을 가지고 있는 부분만을 추출하여 정형화된 형태로 변환하는 것이다. 원하는 정보의 부분만을 추출하기 위하여 임의의 텍스트가 입력으로 주어지며, 사용자가 관심을 가지고 있는 데이터 부분만을 추출하여 출력을 생성한다. 이러한 정보 추출 시스템은 서로 다른 포맷을 사용하는 다양한 정보 소스로부터 특정한 정보 부분을 추출하고 통합하여 일관된 방법으로 사용자에게 제시하기 때문에 사용자의 정보에 대한 만족도를 증가시킬 수 있다.

웹에서의 데이터 추출의 문제를 다루는 보편적인 접근법은 다양한 데이터 소스에 접근하는 이질성을 캡슐화하는 랩퍼(wrapper)를 작성하는 것이다. 랩퍼는 특정한 정보 소스에 대해서 관심있는 데이터의 위치와 구조 포맷 등을 나타내는 추출 규칙이라고 정의할 수 있다. 이러한 랩퍼시스템은 웹 페이지를 유용한 정보들을 포함하고 있는 집합이라고 보고 있다. 전자 상거래에서 상품 검색 결과 문서들이나 부동산 매물 검색 결과 문서와 같이 웹에서 출력되는 준 구조화된 문서들은 문서에서 특정 위치의 데이터 부분만이 중요한 의미를 가지고 있다. 따라서 그 집합 안에서 유용한 정보들을 추출하는 랩퍼 구성 작업은 아주 유용한 일이라고 볼 수 있다. 이와 같은 랩퍼 응용 시스템을 구현함으로써 사용자의 지적 호기심을 충족시켜 줄 수 있는 정보 서비스를 제공할 수 있다. 랩퍼 응용 시스템은 전자상거래 비교 시스템이나 영화 정보를 다양하게 제공하려는 시스템에서 유용하게 사용될 수 있다[27, 30]. 이러한 랩퍼의 추출 성능을 향상시키기 위해서 다음과 같은 요소를 고려해야 한다.

1.1 레이블이 없는 엔티티

정보 소스에서 랩퍼를 생성할 때, 레이블을 가지고 있는 텍스트는 도메인 지식에 의해서 자동으로 인식되게 된다. 그러나 레이블을 가지고 있지 않는 텍스트는 도메인 지식을 이용한다고 하더라도, 해당 텍스트에 대한 의미를 이해할 수 있는 단서가 없기 때문에, 텍스트에 대한 엔티티를 인식할 수가 없게 된다. 이렇게 레이블이 존재하지 않는 엔티티를 인식할 수 있는 방법이 필요하다.

1.2 도메인 지식 확장

도메인 지식 기반 랩퍼 생성 시스템을 구축할 때, 초기 도메인 지식은 도메인 전문가에 의해서 구축된다. 따라서 많은 도메인에 시스템을 적용하기 위해서는 각 분야의 도메인 전문가의 개입이 필수적이다. 그러나 이러한 것은 자동화된 랩퍼 생성을 위해서는 그다지 큰 어려움은 아니다. 초기에 한 번만 구축해 놓으면 계속해서 유용하게 사용할 수 있기 때문이다. 그러나 도메인 지식이 초기에 제대로 구축이 되었다고 하더라도 웹의 빠른 변화와 동적인 특성으로 인해, 해당 도메인에 대한 새로운 특성들이 발견될 수 있다. 이러한 변화를 제대로 수용하지 못한다면 시스템의 도메인 지식은 새로운 데이타들을 수용하지 못하는 결과를 초래하게 될 것이다. 따라서 이미 구축된 도메인 지식과 이것을 이용하여 정보를 추출하는 과정에서, 새롭게 발견되는 해당 도메인의 특성들을 자동 혹은 반자동으로 감지하고 확장할 수 있는 방법의 개발이 필요하다.

1.3 하위 링크의 정보

대부분의 웹 사이트에서는 첫 페이지에 아이템의 간략한 정보만을 제공하고 하이퍼링크를 클릭하여 상세정보를 보고자할 때, 상세 정보를 보여주도록 하고 있다. 이러한 방법은 처음에 많은 정보를 보여줌으로써 발생할 수 있는 시스템의 오버헤드를 줄일 수 있을 뿐만 아니라, 사용자가 정보들을 간략하게 빠른 시간 안에 살펴 볼 수 있도록 한다. 상세 정보는 모든 사람에게 보여주는 것이 아니라

필요한 사람에게만 보여주는 것이다. 따라서 랩퍼 생성 및 정보 추출 시스템은 웹이 가장 크게 발전할 수 있게 되었던 이유 중의 하나인 하이퍼링크의 유용성을 충분히 이용해야만 보다 성능좋은 시스템을 구축할 수 있게 된다.

따라서 본 논문에서는 위의 세가지 고려사항을 만족하는, 구조화된 웹문서에서 자동으로 정보를 추출하고 추출된 정보를 통합하는 웹 정보추출 시스템을 제안한다. 레이블이 없는 엔티티를 인식하기 위해 확률 기반 엔티티인식 방법을 이용하며, 추출된 데이터를 이용하여 기존의 도메인 지식을 반자동으로 확장하는 기능을 제공한다. 추출 대상 기본 페이지에 링크된 하위 링크의 정보를 추출하는 기능을 제공한다. 한 도메인에 대한 이종의 정보 소스로부터 얻어진 유사 추출 결과를 통합하는 기능을 제공한다.

2. 관련연구

랩퍼 생성 시스템은 인터넷 상에 존재하는 다양한 정보 소스에 대해서 정보 추출 규칙을 생성하도록 하는 프로그램이다. 방대한 자료만큼이나 다양한 도메인이 존재하고 있기 때문에, 도메인 지식을 얼마나 잘 활용하여 다양한 웹 사이트에 적용시켜 유용한 정보를 추출할 것인가가 이 프로그램의 핵심 역할이다. 또한 도메인 지식만을 가지고 인식할 수 없는 텍스트에 대해서 확률적 방법을 적용한 엔티티 추출을 시도한다는 점과, 도메인 지식의 표현력이 부족하다고 판단될 때 도메인 지식의 확장을 시도한다는 점은 본 시스템이 가지고 있는 가장 큰 특징이라고 할 수 있다.

도메인 지식을 이용한 랩퍼 기반 정보 추출 시스템은 주어진 URL에 대한 웹문서를 정렬한 후 도메인 지식과 휴리스틱을 이용하여 반복 패턴을 찾아낸 뒤 XML 형태의 문서로 이 패턴을 인식하는 규칙(랩퍼)을 표현하여 저장한다. 그리고 지정 URL에 대한 정보 추출 요구가 있을 경우 시스템은 필요한 경우 질의폼을 분석하여 질의를 생성한 뒤 결과 URL의 웹문서와 생성된 규칙을 읽어 문서에 포함된 정보를 추출한다. 얻어진 추출 결과에 대해 동일한 정보를 포함하는 유사 템플릿일 경우 결과 통합기가 여러 템플릿들을 통합하여 하나의 정보 템플릿을 생성한다. 지정된 URL의 모든 정보를 추출하지 못할 경우 도메인 지식 학습기는 추출 결과들을 이용하여 도메인 지식을 확장한다.

정보추출 규칙인 랩퍼 생성에 대한 연구는 수동생성방법, 반자동 생성방법, 그리고 자동생성방법으로 나뉜다. 수동생성방법[10]은 특정 도메인에서 정보를 추출하기 위한 규칙을 수동으로 생성한다. 수동의 규칙 생성 방법은 사람이 하는 작업이기 때문에 시간이 많이 걸리고 새로운 정보소스에 대해서 확장성과 유연성이 결여되는 문제점이 발생한다.

반자동 랩퍼 정보추출 방법[21]은 수동생성에 따른 문제점을 최소화하기 위해 XWRAP 랩퍼를 학습하기 위해 최소한의 사용자 입력을 받는 방안을 소개하였다. XWRAP은 HTML 문서를 계층 구조의 트리로 구성하고, 의미 부분에 대한 사용자 입력을 받도록 하고 있다. 이 시스템의 문제점은 반드시 사용자 입력을 받아야 하는데 있다. 사용자에게 친근한 인터페이스를 제공한다고 하더라도 시스템의 동작 상황을 정확히 알지 못하는 사용자가 정확한 입력을 준다는 보장은 없다. 또한 계층화된 트리로 구성되지 못하는 HTML 문서의 경우는 학습할 수 없는 문제를 가지고 있다[21].

자동 랩퍼 생성 방법은 크게 기계학습 방법[3-9, 12-20, 22-24, 26, 28, 29], 데이터 마이닝 방법[1, 2, 25], 그리고 개념 모델링 방법[11]으로 나뉜다. 기계학습 방법은 인터넷상의 많은 정보들이 상관관계가 있는 데이터로 존재하고 있다고 보고, 레이블이 포함된 데이터로부터 랩퍼를 자동으로 생성하기 위한 기계학습 기반 랩퍼 귀납법(induction)을 이용한다. 데이터마이닝 방법은 사용자로부터 예제 객체 집합을 수집 및 분석하여 bottom-up extraction에 의해 새로운 웹페이지의 새로운 객체를 추출하는 방법이다. 개념 모델링 방법은 데이터를 추출하고 구조화하기 위해 온톨로지(개념 모델 인스

턴스)를 파싱하여 데이터베이스 스키마를 자동으로 생성하고 키워드를 인식한다. 그후 비구조화 문서에서 데이터를 인식하고 추출하여 생성된 데이터베이스 스키마에 저장한다.

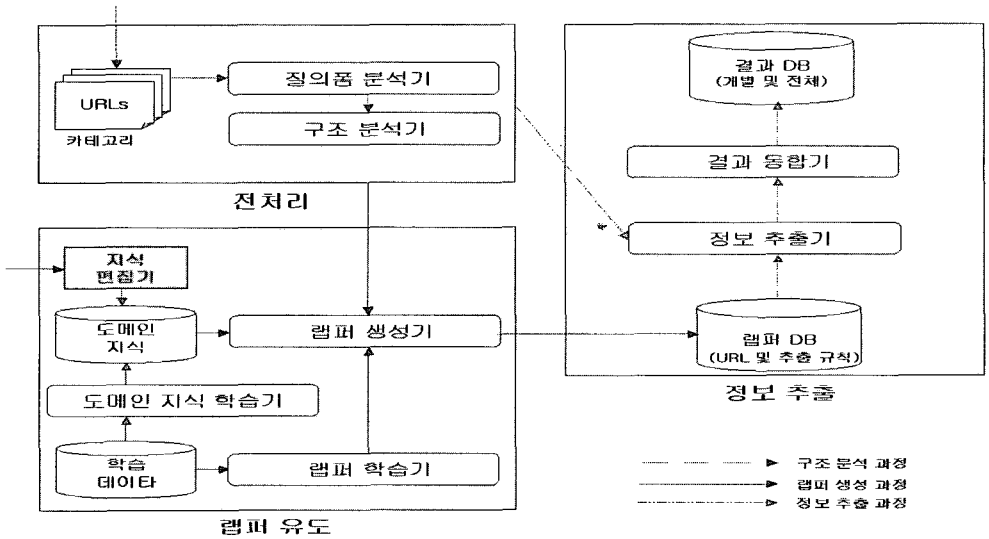
위의 자동 랩퍼 생성 방법은 랩퍼를 기술하기 위한 형식은 다르지만 다음과 같은 공통적인 문제가 있다. 첫째, 도메인 지식과 일치하는 엔티티만을 인식하고, 레이블이 없거나 다른 이름의 레이블이 존재하면 엔티티를 인식하지 못한다. 둘째, 이미 구축된 도메인 지식과 이것을 이용하여 정보를 추출하는 과정에서, 새롭게 발견되는 해당 도메인의 특성들을 자동으로 감지하고 확장할 수 없다. 셋째, 대부분 방법이 첫 페이지에서 정보를 추출할 뿐 상세정보를 포함하고 있는 하이퍼링크를 통한 하위페이지의 정보를 고려하지 않고 있다. 따라서 본 논문에서는 위의 세가지 문제점을 해결하는 기계학습에 기반한 자동 웹 정보추출 방안을 제안한다.

3. 자동 웹 정보추출 방안

랩퍼 생성 시스템의 주요 기능은 도메인 지식의 반자동 학습 기능, 질의폼 분석 및 질의 생성 기능, 하위 링크에 대한 정보 추출 기능, GUI를 통한 시스템 관리 기능, 추출 결과 통합 기능, 확률 기반 제목 추출 기능 등으로 구성된다. 본 논문에서 제안하는 도메인 지식 기반 랩퍼 생성 시스템의 전체적인 구조도는 그림 1과 같다.

정보추출시스템의 랩퍼유도엔진은 학습 데이타를 기반으로 랩퍼를 학습하고, 랩퍼구조에 대한 분석 결과와 도메인 지식을 이용하여 랩퍼생성기가 랩퍼를 생성한다. 이때 도메인 지식만으로 생성되는 랩퍼의 한계를 극복하기 위해 확률 기반의 랩퍼 생성 방법을 추가하도록 한다. 이 방법을 통하여 랩퍼의 추출 성능에 대한 정확도를 높일 수 있다. 랩퍼 생성을 수행한 이후에 사용자의 정보 추출 요구가 있게 되면, 생성된 랩퍼를 이용하여 정보를 추출하도록 한다.

전처리기에서는 전문 쇼핑몰이 증가함에 따라 검색 키워드를 질의하여 각 쇼핑몰의 상품들을 검색할 수 있는 환경이 제공되고 있다. 일반 웹 페이지의 정보를 추출하는 것과는 달리 검색의 결과가 되는 아이템의 리스트는 웹 페이지에 저장되어 있는 내용들이 아니라 데이터베이스 시스템으로부터 사용자 질의 검색에 따라 동적으로 아이템들을 검



〈그림 1〉 정보 추출 시스템 구조도

색하여 웹 페이지를 구성하게 된다. 따라서 쇼핑몰 사이트의 데이터베이스 시스템에 아이템들이 수시로 추가되거나 삭제, 편집될 수 있기 때문에 동일한 질의 검색을 하더라도 다른 웹 페이지가 검색 결과로 나올 수 있다. 이와 같은 환경에서 랩퍼 DB내에 질의 검색을 지원하는 각 사이트마다 질의 표현식을 저장해 놓고 정보 추출기가 정보를 추출하는 시점에서 질의어를 구성하여 정보를 추출하도록 한다.

정보추출엔진은 전처리기에서 질의분석의 결과와 랩퍼유도 엔진에서 생성된 랩퍼를 바탕으로 해당 사이트에 대한 정보를 자동으로 추출한다. 정보추출 시스템의 활용 도메인은 '영화', '음식점', '숙박업소'와 같은 것들이 있을 수 있다. 한 도메인에 대한 정보를 가지는 여러 사이트들로부터 얻어진 추출 결과는 유사한 내용을 포함하고 있어 중복된 정보가 저장될 수 있다. 데이터의 중복을 피하기 위해 이종 사이트로부터 얻어진 정보를 통합할 수 있는 기능을 구현한다. 정보의 통합은 슬롯 단위의 하나 이상의 고유 특징(feature)를 정한 뒤 특징값이 같은 값들에 대해 템플릿들을 통합하여 하나의 통합 템플릿을 구성한다.

3.1 레이블이 없는 엔티티 인식 방안

정보 소스에서 랩퍼를 생성할 때, 레이블을 가지고 있는 텍스트는 도메인 지식에 의해서 자동으로 인식되게 된다. 그러나 레이블을 가지고 있지 않는 텍스트는 도메인 지식을 이용한다고 하더라도, 해당 텍스트에 대한 의미를 이해할 수 있는 단서가 없기 때문에, 텍스트에 대한 엔티티를 인식할 수가 없게 된다. 본 논문에서는 이렇게 인식되지 않는 텍스트의 의미를 이해하기 위해서 확률적인 방법을 새롭게 도입하고자 한다.

우선 모델을 제안하기 이전에 관련된 용어에 대해서 정의한다. '엔티티'는 도메인에서 유용하게 사용될 수 있는 구성 요소의 기본 단위이다. 예를 들어, 제목이나 감독, 혹은 주연과 같은 정보들이 엔티티가 될 수 있다. '레이블'은 해당 정보 소스에서 엔티티를 인식할 수 있도록 제공하는 단서이다. 제목이라는 엔티티를 표현하기 위해서 정보 소스는 제목이라고 레이블을 줄 수도 있겠지만, 타이틀이라고 줄 수도 있고, 영화 제목이라고 줄 수도 있다. 즉, 레이블은 엔티티와 같은 의미로 사용되는 유사어라고 볼 수 있다. '아이템'은 정보 소스에서 제공하는 정보의 기본 단위라고 정의할 수 있다. 대부분의 웹 정보 소스가 페이지에 여러 아이템을 일정 패턴(리스트 형태나 테이블 형태)에 맞게 표시를 하고 있다. 이러한 정보들은 대부분이 데이터베이스로 구축되어 있는 것들이고, 해당 웹 프로그램이 데이터베이스에 접근하여 반복적으로 아이템들에 대한 정보를 생성한다. 따라서 아이템은 데이터베이스의 투플이라고도 정의될 수 있다. 웹 문서에 대한 구조 분석을 수행할 경우에 텍스트 조각들이 태그에 의해서 띄엄 띄엄 떨어져서 나오게 되는데, 이러한 텍스트 조각들을 브라우져에서 보여지는 것과 같이 논리적으로 묶어서 의미를 가질 수 있는 텍스트로 재구성하게 된다. 이렇게 구성된 텍스트에서 엔티티의 값이 될 수 있는 부분을 토큰이라고 칭한다. 구조 분석을 수행하면 많은 토큰들이 생기게 된다.

여러 아이템의 정보를 담고 있는 페이지에서 하나의 아이템을 기준으로 살펴보면, 레이블이 있는 텍스트와 그렇지 않은 텍스트가 있다. 레이블이 있는 텍스트는 도메인 지식을 이용하여 의미 정보와 구조 정보를 자동으로 인식해 낼 수 있다. 그러나 레이블이 없는 경우 텍스트의 의미를 자동으로 알아내기 위해서는 확률적인 방법 등을 적용하여야 한다. 구조 분석을 수행하면 레이블이 있는 정보와 레이블이 없는 정보가 정보 소스에서 제공하는 모든 아이템에 대해서 같은 패턴을 가지고 나오기 때문에, 정보 소스에 대해서 토큰 집합이라는 것을 구성할 수 있다. 즉, 하나의 아이템에 대해서 토큰을 하나 선택하면, 이러한 토큰들을 모아서 구성한 것을 토큰 집합(token set)이라고 말할 수 있다. 따라서 하나의 정보 소스에서 여러 개의 토큰 집합

을 구성할 수 있게 된다. 여러 개의 토큰 집합이 순차적으로 나오기 때문에 이것을 토큰 집합 열(token set sequence)이라고 부르도록 하겠다. 위와 같은 내용을 수학적으로 정리한다.

1) 하나의 아이템에 대해서 n개의 인식된 토큰 $\{t_1 t_2,..., t_n\}$이 있다.
2) n개의 할당된 엔티티 $\{e_1 e_2,..., e_n\}$가 있다.
3) 하나의 아이템에 대해서 m개의 인식되지 않은 토큰 $\{t_1 t_2,..., t_m\}$이 있다.
4) q개의 할당되지 않은 엔티티 $\{e_1 e_2,..., e_q\}$가 있다.

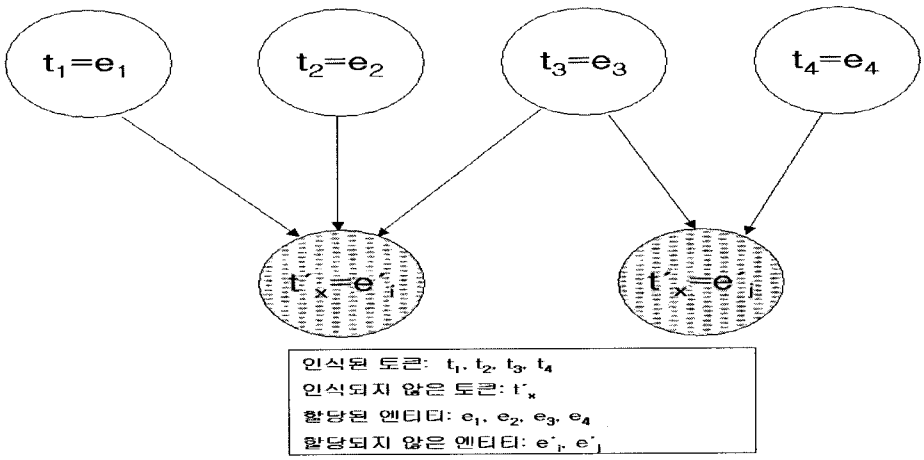
이때 $e'k$는 도메인 지식에서 정의된 엔티티 집합 E에서 현재의 정보 소스에서 발견된 엔티티를 뺀 나머지 집합이다. 토큰에 대한 엔티티는 배타적으로 부여되기 때문에 이미 발견된 엔티티는 새롭게 인식될 수 있는 엔티티 집합에서 제거해야만 한다.

5) 하나의 정보 소스에 대해서 V개의 아이템이 존재한다.
6) 하나의 정보 소스에 대해서 n개의 인식된 토큰 집합 $\{t_1, t_2,..., t_n\}$이 있다. 그리고 하나의 토큰 집합에는 V개의 토큰 이 있다. $T = i\{t_{i1}, t_{i2},..., t_{iv}\}$
7) 하나의 정보 소스에 대해서 m개의 인식되지 않은 토큰 집합 $\{t_1 t_2,..., t_m\}$이 있다. 그리고 하나의 토큰 집합에는 V개의 토큰 $T = i\{t_{i1}, t_{i2},..., t_{iv}\}$이 있다.
8) 도메인 지식에 ($n+m$)개의 엔티티가 있다.

위에서 정의된 것을 바탕으로 토큰 집합에 엔티티 이름을 배타적으로 부여하는 확률 정보 기반 모델에 대해서 제안하고자 한다. 하나의 아이템 안에 같이 존재하는 주변 정보들을 이용하는 방법으로서, 토큰과 같은 아이템에 속해 있는 레이블이 있는 텍스트 정보를 이용하는 방법이다. 도메인 지식에 의해서 이미 인식된 텍스트 정보를 이용하면, 인식되지 않은 토큰의 레이블을 추정해 볼 수 있기 때문이다. 이것은 기존에 추출되었던 아이템들이 관련 데이터를 가지고 있기 때문에 적용가능하다. 즉, 여러 정보 소스에 대해서 랩퍼를 생성하고 정보를 추출하도록 하고 있기 때문에, 다른 정보 소스에서 추출된 정보를 이용하여 현재 사이트에서 문제가 되고 있는 것들을 해결할 수 있다.

예를 들어 '취화선'이라는 영화의 제목이 레이블이 없어서 인식이 되지 않았다고 가정해 보자. '취화선'은 감독이 '임권택'이고, 주연이 '최민식'인 영화의 제목이다. 이때, 텍스트에 레이블이 있어서 이미 '임권택'이라는 토큰이 감독 엔티티로 인식이 되었을 경우에는 '취화선'이라는 토큰이 제목 엔티티가 될 확률이 많아진다. 왜냐하면 다른 정보 소스에서 이미 {(제목='취화선'), (감독='임권택'), (주연='최민식')}이라는 데이터를 가진 아이템을 추출해서 학습 데이터에 추가시켰을 가능성이 있기 때문이다. 또한 여기에 주연이 '최민식'이라는 토큰을 가지고 있다면 '취화선'이라는 토큰은 더욱 더 제목 엔티티가 될 확률이 많아진다. 만약에 기존에 취화선이라는 제작사가 있어서, '임권택' 감독에게 영화 제작을 맡겼다면, {(감독='임권택'), (제작='취화선')}이라는 아이템이 있을 수 있다. 위와 같이 학습 데이터에 존재한다면 (감독='임권택')이라는 정보만을 가지고서는 '취화선'이라는 토큰이 제작사 엔티티인지, 제목 엔티티인지를 알 수 없게 된다. 이러한 경우에 있어서 (주연='최민식')이라는 데이터는 '취화선'이라는 토큰이 제목이라는 엔티티로 갈 확률을 더욱 크게 만드는 역할을 한다.

위와 같이 컨텍스트 정보는 인식되지 않은 토큰의 엔티티를 식별하는데 있어서 중요한 역할을 할 수 있다. 이러한 확률 값은 기존에 추출되었던 데이터에 의해서 계산될 수 있다. 단, 이때 하나의 아이템에 대한 컨텍스트 정보만을 고려하는 것이 아니라, 여러 개의 아이템이 존재하기 때문에, 여러 아이템에 대한 컨텍스트 정보를 고려하도록 한다. 하나의 아이템에 대한 컨텍스트 정보를 이용하는 것보다는 여러 개의 아이템에 대한 컨텍스트 정보를 이용하는 것이 좀 더 변별력있는 확률 값



〈그림 2〉 인식된 토큰과 인식되지 않은 토큰의 관계

을 구할 수 있기 때문이다. 이러한 개념을 이용하면 정보 소스의 아이템에 대해서 레이블이 없어서 식별되지 않는 토큰들을 확률 값을 이용하여 새로운 엔티티로 식별할 수 있게 된다.

그림 2에서 보여지는 것과 같이 컨텍스트 정보를 이용하여, 즉 인식된 토큰의 엔티티 정보를 이용하여 인식되지 않은 토큰의 엔티티를 식별하기 위한 수식을 모델링할 수 있다. 이 모델에서는 토큰 t'_X가 엔티티 e'_i로 인식될 것인지, 엔티티 e'_j로 인식될 것인지를 알기 위해서 이미 밝혀진 토큰 정보 $\{(t_1 = e_1),(t_2 = e_2),(t_3 = e_3),(t_4 = e_4)$를 이용하는 것이다. 즉, 기존의 통계 데이터에 t_1가 e_1이고 t'_X가 e'_i인 경우, t_2가 e_2고 t'_X가 e'_i인 경우, t_3가 e_3이고 t'_X가 e'_i인 경우가 t_3가 e_3이고 t'_X가 e'_j인 경우, t_4가 e_4이고 t'_X가 e'_j인 경우보다 많으면 t'_X는 e'_j보다는 e'_i일 가능성이 훨씬 클 것이다. 따라서 그림 2에서 화살표의 의미는 양쪽의 노드가 같이 존재할 수 있는 경우의 확률 이라고 볼 수 있다. 화살표의 확률이 클수록, 그리고 화살표가 많이 연결되어 있을수록 해당 노드의 할당되지 않은 엔티티가 새롭게 할당될 확률은 더욱 커지게 된다. 이러한 개념을 이용해서 이미 밝혀진 노드 정보가 새롭게 인식될 노드 정보를 지원(support)하는 정도를 수식으로 표현해 볼 수 있다.

다음과 같은 과정을 거쳐서 모델의 확률 값을 계산할 수 있다.

1) 여러 개의 정보 소스로부터 학습 데이터를 구축한다.
2) 정보 추출을 수행할 정보 소스에 대해서, 전 제에서 제시한 데이터들을 구성한다.
3) 토큰이 엔티티에 속할 확률 값을 계산한다.

$t_1 = e_1, t_2 = e_2, t_3 = e_3, t_4 = e_4$가 인식되었을 때 임의의 토큰 t'_j가 엔티티 e'_i일 확률은 식 (1)과 같다.

$$P(t'_j = e'_i | \{t_1 = e_1, t_2 = e_2, \ldots t_n = e_n\}) \quad (1)$$

전확률(Total probability)에 의해서 위의 식(1)은 다음과 같은 식 (2)로 변형된다.

$$\sum_{k=1}^{n} P(t'_j = e'_i | \{t_1 = e_1, t_2 = e_2, \ldots t_n = e_n\}) = \sum_{k=1}^{n} P(t'_j = e'_i, t_k = e_k) \quad (2)$$

식 (2)에 의해서 임의의 토큰 집합 $T'_j, j = 1, \ldots n$는 $T_1 = e_1, T_2 = e_2, T_3 = e_3, T_4 = e_4$가 인식되었을 때 엔티티 e'_i일 확률은 식 (3)과 같다.

$$P(T'_j = e'_i | \{T_1 = e_1, T_2 = e_2, T_3 = e_3, T_4 = e_4,\})$$
$$\cong \frac{1}{v}\sum_{h=1}^{u}\sum_{k=1}^{n} P(T'_j = e'_i | \{T_1 = e_1, T_2 = e_2, T_3 = e_3, T_4 = e_4,\}) \quad (3)$$

그러나 정보 소스에 여러 개의 아이템이 존재하기 때문에, 토큰이 엔티티에 속할 확률 값보다는 토큰 집합이 엔티티에 속할 확률 값을 계산하는 것이 보다 신뢰성있는 정보를 얻을 수 있다. 따라서 식 (3)과 같이 계산하도록 한다.

3.1) $P(e_i = t_{jk} | e_h = t_{hk})$는 학습 데이터로부터 얻어지며, 이 확률은 엔티티 e'_i 값이 t'_{jk}이고, 엔티티 e_h값이 t_{hk}인 투플의 개수를 학습 데이터의 전체 투풀 개수로 나눈 값이다.

3.2) $P(e_h = t_{hk})$는 학습 데이터로부터 얻어지며, 이 확률은 엔티티 e_h값이 t_{hk}인 투플의 개수를 학습 데이터의 전체 투플의 개수로 나눈 값이다.

4) $P(T_j = e_i | \{T_1 = e_1, T_2 = e_2, T_3 = e_3, T_4 = e_4,\})$가 가장 큰 확률 값을 갖는 e'_i를 선택하여, 토큰 집합 T'_j의 엔티티로 할당한다. 단, 이때 토큰이 엔티티가 될 확률이 임계 값을 넘지 않을 경우에는 해당 토큰의 엔티티 식별은 무효로 한다.

5) 처음의 토큰 집합 열로부터 토큰 집합 T'_j를 제거하여, 새로운 토큰 집합 열 $T_1, T_2, \ldots, T_{m-1}$을 생성한다. 새롭게 생성된 토큰 집합 열에 대해서 단계 3)과 4)를 반복해서 적용한다.

모델을 적용시켜서 확률 값을 계산하는 방법은 다음과 같다.

1) 학습 데이터를 구축한다. 학습 데이터에 u개의 투플이 존재한다고 가정한다.

2) 확률 값을 계산한다.

$$P(t'_1 = e'_2 | \{t_4 = e_4, t_5 = e_5, t_6 = e_6\})$$
$$= P(t'_1 = e'_2, t_4 = e_4) + P(t'_1 = e'_2, t_5 = e_5) + P(t'_1 = e'_2, t_6 = e_6)$$
$$= \frac{\#ofitem(\ e'_2 = t'_1\ \&\ t_4 = e_4)}{u} * \frac{\#ofitem(\ t_4 = e_4\ )}{u} + \quad (4)$$

정보 소스에서 인식되지 않은 토큰 t'_1이 엔티티 e'_2에 속할 확률 값은 식 (4)와 같이 계산된다. 이때 레이블이 있는 텍스트 t_4는 e_4로, t_5는 e_5로, t_6은 e_6으로 이미 인식이 되었다고 가정한다. 여기에서 u는 학습 데이터에 존재하는 투플의 개수이고, $\#ofitem$은 임의의 i에 대하여 $t_1 = e_2$과 $t_1 = e_2$를 동시에 만족하는 아이템의 개수이다.

3.2. 반자동 도메인 지식의 확장 방안

반자동 도메인 지식의 확장이란 초기에 수동으로 작성된 도메인 지식을 확장하기 위해 자동으로 확장 가능 후보를 추출한 다음 수작업으로 최종 결정하여 선택하는 반자동 확장 방법을 말한다. 랩퍼 생성은 초기의 도메인 전문가에 의해서 구축된 도메인 지식에 기반하여 이루어지도록 되어 있다. 그러나 랩퍼를 생성할 적절한 도메인 지식 항목을 찾을 수 없을 경우에는 대상 사이트에 대한 랩퍼를 생성할 수 없게 된다. 이는 해당 사이트에서 제공되는 레이블이 도메인 지식의 용어(term)에서 찾을 수 없거나, 혹은 찾을 수 있더라도 도메인 지식에 정의된 포멧의 형태, 혹은 구분자(delimiter)의 종류가 달라서 인식할 수 없는 경우로 볼 수 있다. 따라서 정보 소스의 구조적 변화, 내용적 변화에 대처하기 위해 초기에 작성된 도메인 지식을 능동적으로 변경하여 확장하는 도메인 지식 학습 방법이 필요하다.

랩퍼 생성에 실패하였을 경우 최근의 추출 데이터를 이용하여 현재 사이트의 구조를 인식할 수 있는 랩퍼를 생성할 수 있게 하기 위해 도메인 지

식의 확장을 시도한다. 이전에 추출된 결과들의 레이블과 구분자는 반드시 도메인 지식에 포함된 값(value) 중의 하나가 될 것이다. 때문에 과거 추출 결과를 이용하여 새로운 레이블과 구분자를 식별하는 것은 불가능하다. 그러나 값은 일정한 형식을 지니는 여러 값들을 가질 수 있으므로 값들을 이용하여 레이블과 구분자를 추론하도록 한다.

도메인 지식 학습을 위해 다음을 가정한다.

- 정보의 기본 단위인 슬롯은 레이블, 구분자, 값으로 구성된다. 템플릿은 슬롯의 집합이다.
- 슬롯의 구성정보에는 값의 형식을 나타내는 value_type, 구성 요소들 간의 나열 정보를 나타내는 프로퍼티(property)가 있다.
- 구분자로 사용될 수 있는 특수 문자 정보를 이용하여 텍스트 토큰을 레이블, 구분자, 값으로 나눈다. 구분자는 주로 심벌(symbol)로 구성되므로 심벌데이터, 즉 특수 문자 정보를 구분자 후보로 인식하도록 하여 텍스트를 여러 가지 경우로 분리해 볼 수 있다.
- 각 슬롯의 값에 나타나는 값들을 학습 데이터를 이용하여 어떤 엔티티로 인식하는 것이 가장 적당한가를 결정하도록 한다. 여기서 엔티티가 결정되면, 레이블은 엔티티의 레이블로 결정되고, 구분자는 엔티티를 인식할 수 있는 구분자로 등록이 된다.

우선 랩퍼 생성에 실패한 사이트에 대해 사이트 구조 분석을 하여 객체 트리를 생성한다. 객체가 식별되면 도메인 지식을 구성할 수 있는 후보들을 찾기 위해 우선 값 후보들을 선택한다. value가 결정되면 레이블과 구분자를 결정하고 이것들을 이용하여 확률 값을 계산한다. 계산된 확률 값을 최대 값으로 갖는 요소를 이용하여 엔티티, 레이블, 프로퍼티를 결정한다. 여기서 결정된 요소를 도메인 지식의 적절한 부분에 추가시키고, 다시 랩퍼를 생성하도록 한다. 도메인 지식에서 부족한 표현력을 학습 데이터를 이용하여 확장하였기 때문에 새롭게 랩퍼를 생성할 경우에 제대로 된 랩퍼를 얻을 수 있는 가능성이 커진다. 아래의 과정을 템플릿을 구성하는 모든 슬롯에 대해 반복한다.

- **값의 인식**

  기존의 통계 데이터와 목표 사이트의 데이터를 비교한다. 우선 타입을 비교하여 value_ type이 같을 경우 같은 슬롯만을 대상으로 확률 값을 계산한다. 타입이 같을 경우 과거에 추출된 데이터의 값에 나타난 단어집합과 목표 사이트에 나타난 후보 객체들의 단어집합을 비교하여 겹치는 데이터가 많을수록 해당 슬롯의 값이 될 확률이 높다. 추출된 데이터의 단어 벡터와 후보 객체의 단어 벡터간의 벡터 유사도를 계산하여 유사도가 높은 슬롯으로 결정한다.

- **레이블의 인식**

  값을 인식하여 슬롯이 결정되면 나머지 후보 객체들 중에서 레이블을 선택한다. 레이블은 도메인 지식에 등록되지 않은 전혀 다른 값이거나 등록된 값과 심벌 등이 조합된 값일 수 있다.

- **구분자의 인식**

  값과 레이블이 선택되면 구분자를 결정할 수 있다.

위와 같은 과정을 거쳐서 가장 유용하다고 판단된 엔티티, 레이블, 구분자를 도메인 지식의 각 항목에 확장한다.

3.3. 하위링크의 정보 통합 방안

하위링크의 정보통합이란 현재의 페이지에 대한 정보가 충분하지 못하다고 판단되면, 하위링크로 연결된 페이지의 정보까지 추출하여 현재의 페이지의 정보추출 결과와 하위링크로 연결된 페이지의 정보추출 결과를 통합하는 것을 말한다.

많은 웹 정보 소스가 사용자에게 정보를 제공할 때, 처음에는 간략 정보만을 제공하는 방식을 취하고 있다. 해당 아이템의 상세 정보를 보기를 원했

을 경우에만 하이퍼링크로 연결되어 있는 상세 정보를 보여주도록 한다. 이것은 사용자에게 처음에는 아이템의 중요 정보만을 제공하고, 추가 정보를 원할 경우에만 상세 정보를 보도록 하기 때문에, 자신이 원하는 정보를 대략적으로 빨리 훑어볼 수 있는 장점이 있다. 처음부터 모든 정보를 제공하는 것은 사용자에게 원하지 않는 정보를 제공하여 불편하도록 할 수 있을 뿐만 아니라, 정보 소스에서 제공하는 데이터를 전체적으로 살펴보는데 있어서 많은 불편함을 줄 수 있다. 또한 사용자가 처음에 접속하는 페이지에 많은 정보를 주기 위해서는 데이터베이스에서 한 번에 모든 정보를 가져와서 웹 페이지를 생성해야 하는데, 이럴 경우에 정보를 생성하도록 하는 웹 프로그램의 동작 시간이 상당히 길어질 수 있다. 이것은 사용자가 정보 소스에 접속할 때 초기 접속 시간을 길어지게 하기 때문에, 사용자에게 서비스에 대한 불편을 초래할 수 있다.

따라서 대부분의 웹 정보 소스가 처음에는 중요한 간략 정보만을 제공하고, 추가 정보를 원하는 사람의 경우에 한해서만 상세 정보를 보도록 하는 방식을 취하고 있다. 이렇게 단계별로 원하는 정보를 찾아 들어갈 수 있도록 해주는 하이퍼링크 매커니즘은 웹이 성장할 수 있었던 가장 큰 이유로, 사용자에게 정보를 편리하게 제공하는 방식이다. 따라서 아이템에 대한 충분한 정보를 얻기 위해서는 이러한 하이퍼링크에 연결되어 있는 정보를 잘 활용하여야 한다. 본 논문에서는 이러한 하이퍼링크 정보를 이용하여 정보 소스에서 제공하는 아이템의 상세 정보도 추출하도록 한다.

웹 페이지에는 수많은 하이퍼링크가 존재하고 있다. 따라서 랩퍼를 생성할 때에 아이템에 연결되어 있는 여러 개의 하이퍼링크 중에서 상세 정보를 담고 있는 유용한 하이퍼링크가 어떤 것인지를 알아내야 한다. 그래야만 나중에 정보 추출기가 중요한 정보를 가지고 있는 하이퍼링크에만 구조 분석을 수행하여, 정보를 빠르게 추출할 수 있기 때문이다. 하이퍼링크를 이용하기 위한 방법은 다음과 같다.

● 랩퍼 생성시
- 첫 페이지에서 제공되는 정보들의 패턴을 분석하여 각 아이템의 경계를 감지한다.
- 감지된 바운더리 안에 있는 모든 하이퍼링크를 따라가서 유용한 정보가 있는 지를 확인한다. 도메인 지식을 이용해서 인식된 엔티티의 개수가 가장 많은 문서가 유용한 문서이다.
- 하이퍼링크의 정보가 유용하다고 판단되면 링크의 위치와 발견된 엔티티 관련 정보를 통합하여 랩퍼에 기록한다.

● 정보 추출시
- 랩퍼를 읽어 들여 하이퍼링크에서 정보를 추출해야 하는 지를 결정한다.
- 처음 페이지에서 정보를 추출하고, 하이퍼링크의 정보 추출 표시가 있으면 하이퍼링크에 연결된 페이에서도 정보를 추출한다.
- 첫 페이지(Front page)에서 정보를 추출한 것과 하이퍼링크에 연결된 페이지(Back end page)에서 정보를 추출한 것을 하나의 아이템 단위로 합쳐서 통합된 추출 정보를 생성한다.

이와 같이 하이퍼링크에 포함되어 있는 정보를 분석해서 이용함으로써, 정보 소스에서 얻을 수 있는 유용한 엔티티의 개수를 증가시킬 수 있다.

4. 실험

4.1. 실험 데이터

본 논문에서는 초기에 학습 데이터를 미리 구축해 놓는 방법에서 탈피하여 정보 소스 별로 도메인을 구분하여 놓은 리스트를 입력으로 받아서 배치 처리 방식으로 랩퍼를 생성한 후에, 생성된 랩퍼를 이용하여 단계적으로 학습 데이터를 구축하도록 한다. 초기에는 학습 데이터가 구축되어 있지 않기 때문에 랩퍼 학습을 수행하더라도, 불완전한 랩퍼가 생성될 수 있다. 이것은 초기의 학습 데이터가 불완전하기 때문이다. 그러나 초기에 랩퍼를

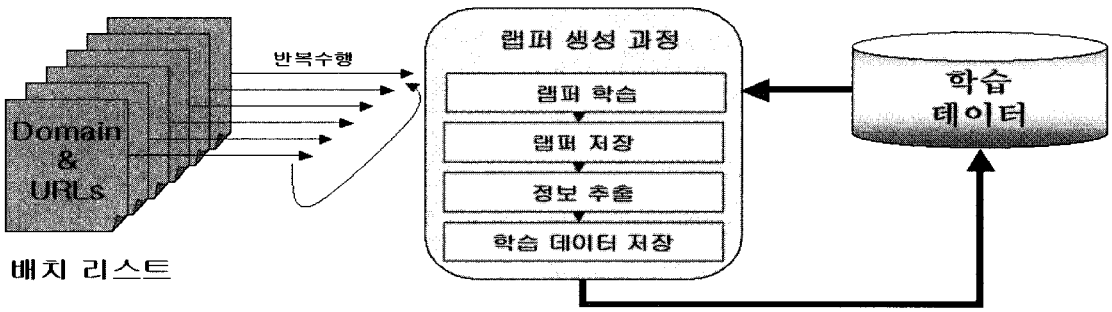
〈표 1〉 실험에 사용된 웹 사이트

도메인	이름	사이트 URL
영화	site	http://www.corecine.co.kr/movie/list_cinecore.htm
영화	site	http://www.joycine.com/omni/time.asp
영화	site	http://www.maxmovie.com/join/cineplex/default.as
영화	site	http://www.maxmovie.com/movieinfo/reserve/movieinfo_reserve.asp
영화	site	http://www.nkino.com/moviedom/coming_movie.as
영화	site	http://www.ticketpark.com/Main/MovieSearch.asp
영화	site	http://www.yesticket.co.kr/ticketmall/resv/movie_main.as

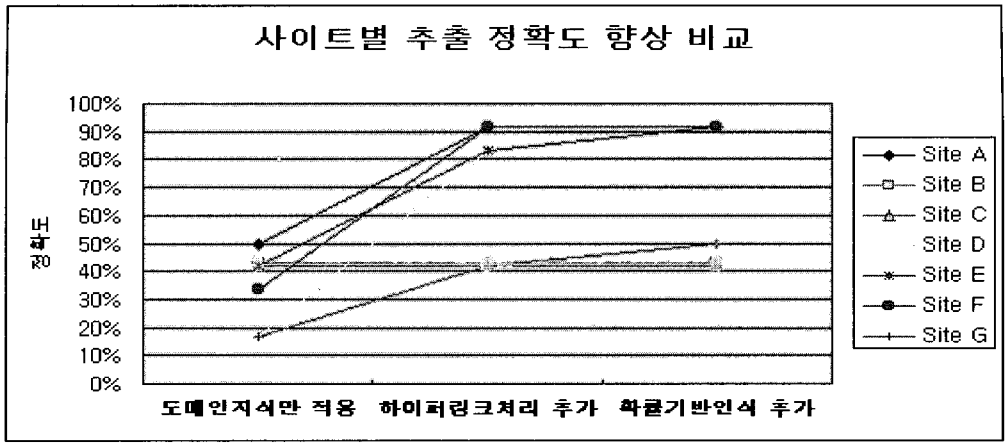
학습하는 과정에서 생성된 랩퍼를 가지고, 정보 추출을 수행하여 학습 데이터를 확장하도록 하고 있기 때문에, 랩퍼 학습을 반복해서 수행할수록 좀 더 정확한 랩퍼를 생성할 수 있게 된다. 이러한 이유는 서로 다른 정보 소스에서 추출된 데이터들이 임의의 정보 소스에 대한 랩퍼를 생성할 때, 확률 값을 계산할 수 있는 근거가 되는 데이터가 돼서 확률 값을 증가시키기 때문이다. 처음에 랩퍼를 생성할 때는 확률 값을 증가시키는 데이터가 없어서 확률 값이 임계 값을 넘지 못했을 수 있다. 그러나, 랩퍼 생성과 생성된 랩퍼를 이용한 정보 추출을 반복할 수록, 확률 값을 증가시킬 수 있는 데이터들이 다른 정보 소스에서 조금씩 생겨나기 때문에, 임의의 랩퍼 생성에 대한 확률 값이 점차 증가하는 효과를 얻을 수 있다. 실제 실험 결과, 처음 랩퍼를 생성할 때는 확률 값을 이용한 엔티티 인식이 실패하였으나, 2~3번 했을 경우에는 레이블이 없는 토큰이 제목과 같은 엔티티로 새롭게 인식되는 것을 관찰할 수 있었다.

학습 데이터를 구축하는 과정은 결국 그림 3과 같이 나타낼 수 있다.

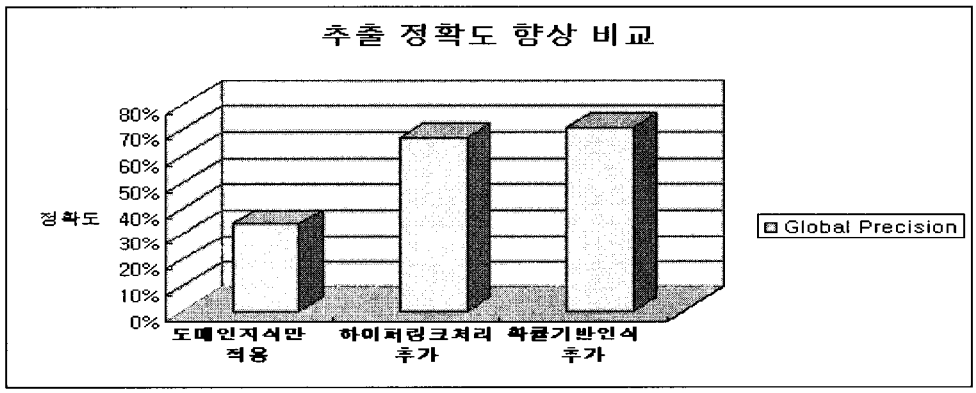
본 논문에서 제안한 랩퍼 생성 시스템을 영화 도메인의 정보 소스에 적용시켜 보았다. 영화에 관련된 도메인 지식을 구축할 때 시스템의 응용 분야에 맞게 도메인 지식의 엔티티를 적절히 선택하도록 해야 한다. 본 논문에서는 도메인 확률 기반 랩퍼 생성 시스템에 대한 평가를 수행하기 위해서 영화 도메인에서 생각해 볼 수 있는 모든 엔티티를 포함하도록 하여, 영화에 관련된 최대 도메인 지식을 가지고 실험을 하였다. 본 논문에서는 영화 도메인에 속한 7개의 웹 정보 소스에 대해서 배치 파일을 구성였다. 이렇게 구성된 배치 파일의 정보



<그림 3> 랩퍼 생성 및 학습 데이터 구축 과정



〈그림 4〉 사이트별 추출 정확도 향상 비교



〈그림 5〉 전체적인 추출 정확도 향상 비교

소스에 대해서 랩퍼 학습과 랩퍼 생성을 반복하면서 서서히 증가하는(incremental) 학습 데이터를 구축할 수 있게 된다. 실험에 사용된 7개의 웹 정보 소스는 표 1과 같다.

본 논문에서 정의한 영화 도메인의 엔티티는 제목, 장르, 감독, 출연, 등급, 제작, 각본, 촬영, 음악, 상영시간, 시작일 그리고 종료일로 이루어져 있다. 그러나 실제 응용 시스템에서는 각본이나 촬영 그리고 음악과 같은 엔티티는 별로 중요하게 취급되지 않을 수 있다. 또한 예매하고는 상관없이 단순히 영화에 대한 정보 제공이 목적이라면, 시작일이나 종료일과 같은 엔티티도 중요하게 취급되지 않을 것이다. 따라서 실제 응용 시스템에서는 본 논문에서 실험한 도메인 지식의 일부 집합만을 적용해도 충분히 실용 가치가 있을 것이다.

4.2. 실험 및 결과 분석

본 논문에서 제안한 몇 가지 방법들의 유용성을 검증하기 위해서 실험은 단계적으로 수행하도록 하였다. 즉, 처음에는 도메인 지식만을 적용하여 랩퍼를 생성하도록 하였고, 다음에는 하이퍼링크에 대한 처리를 추가하여 랩퍼를 생성하도록 하였다. 대부분의 웹 정보 소스가 첫 페이지에는 컨텐트의

간략적인 설명만을 제공하고, 하이퍼링크로 연결된 다른 페이지에 실제로 해당 아이템의 상세한 설명을 제공하는 방식을 취하고 있다. 따라서 하이퍼링크까지 고려하여 랩퍼를 생성해 보면, 좀 더 좋은 결과를 만들어 낼 수 있을 것이다. 마지막으로 본 논문에서 가장 중요하게 생각하는 인식되지 않은 토큰들에 대한 엔티티 인식 알고리즘을 적용하여 랩퍼를 생성하여 그 결과를 비교하였다. 각 사이트의 추출 성능의 정확도는 다음과 같이 계산된다.

$$\text{정확도(precision)} = \left(\frac{\text{추출된 엔티티의 개수}}{\text{추출해야될 엔티티의 개수}}\right) \times 100$$

여기서 추출된 엔티티의 개수는 랩퍼를 학습하면서 인식된 엔티티의 개수라고 볼 수 있고, 추출해야 될 엔티티의 개수는 영화 도메인에서 정의한 엔티티의 개수라고 볼 수 있다. 본 논문에서는 영화 도메인에 12개의 엔티티를 정의하였다. 따라서 추출해야 될 엔티티의 개수는 12가 된다. 전체 사이트에 대한 평균 정확도는 다음과 같이 계산된다.

$$\text{평균정확도 (Global precision)} = \left(\frac{\text{각 사이트의 정확도}}{\text{평가를 수행한 사이트의 개수}}\right)$$

본 논문에서는 영화 도메인에 속하는 7개의 웹 정보 소스를 가지고 테스트를 수행하였기 때문에, 평가를 수행한 사이트의 수는 7이 된다. 사이트별 추출 성능 향상에 대한 결과는 그림 4과 같다. 전체적인 추출 성능 향상에 대한 결과는 그림 5과 같다.

처음 실험에서는 도메인 지식만을 적용하여 랩퍼를 생성하도록 하였다. 실험 결과, 해당 정보 소스에서 추출할 수 있는 엔티티들에 대해서 적절하게 랩퍼를 생성하는 것을 관찰할 수 있었다. 그러나 이러한 방법은 웹 사이트가 가지고 있는 하이퍼링크의 유용성을 제대로 활용하지 못한 결과를 초래하였다. 즉, 대부분의 웹 사이트에서는 첫 페이지에 사용자가 전체적으로 간략하게 살펴볼 수 있는 컨텐트의 간략적인 설명만을 제공하고, 하이퍼링크로 연결된 다른 페이지에 실제로 해당 아이템의 상세한 설명을 하도록 해놨는데, 이러한 특성들을 전혀 이용하지 않고 있었던 것이다. 따라서 추출할 수 있는 엔티티의 수에 많은 제약이 있다고 볼 수 있다.

두 번째 실험에서는 하이퍼링크에 대한 처리를 수행하여 랩퍼를 생성하도록 하였다. 실험 결과 일부 정보 소스에서 추출할 수 있는 엔티티의 수가 배가 넘게 증가하는 것을 관찰할 수 있었다. 이것은 웹 사이트의 구조적 특성을 감안하여 하이퍼링크에 대한 처리를 수행했기 때문이라고 보여진다. 웹을 필두로 한 인터넷의 발전에 가장 크게 기여한 요소가 하이퍼링크라고 말하는 경우가 많은데, 실제적으로 웹 정보 소스를 기반으로 한 랩퍼 생성 시스템에서도 이러한 하이퍼링크의 특성을 이용하는 것이 효과적임을 살펴볼 수 있었다.

세 번째 실험에서는 인식되지 않은 토큰들에 대해서 엔티티 인식 알고리즘을 적용하여 랩퍼를 생성하도록 하였다. 실험 결과 일부 정보 소스에서 추출할 수 있는 엔티티의 수가 증가하는 것을 관찰할 수 있었다. 이것은 레이블이 없는 토큰들에 대해서 확률적 방법을 적용해서 엔티티 인식을 수행한 방법이 효과가 있었다는 것을 보여준다. 이 결과에서 새롭게 인식된 엔티티의 성격을 살펴 볼 필요가 있다. 타이틀과 같은 정보는 어느 도메인에서 사용되든지 간에 항상 존재해야만 하는 핵심 엔티티라고 볼 수 있는데, 이러한 정보들이 추출되지 않으면 정보 추출은 자칫 무의미한 작업이 될 수도 있다. 그러나 많은 정보 소스에서 타이틀과 같은 중요한 정보에 레이블을 주지 않는 경우가 상당수 발견되고 있다. 이것은 타이틀과 같이 중요한 정보에 대해서는 텍스트의 폰트를 키우거나 색깔을 화려하게 부각시켜서 가장 중심적인 내용이라는 것을 알려주려고 하기 때문이다. 그리고 타이틀과 같이 아이템을 구별하는 정보로 사용되는 엔티티는 사용자의 직관에 의해 쉽게 인지될 수 있기 때문에 레이블을 붙이지 않는 이유도 있다고 보여진다. 이러한 문제를 해결하기 위해서 확률 기

〈표 2〉 다른 시스템과의 비교

	구조문서	자연어문서	다중슬롯	레이블없는 경우	하위링크 고려	지식 확장 방법
ShopBot	O	X	처리불가능	추출불가	미고려	수동
WIEN	O	X	처리가능	추출불가	미고려	수동
SoftMealy	O	X	처리불가능	추출불가	미고려	수동
STALKER	O	X	처리가능	추출불가	미고려	수동
RAPIER	O	X	처리불가능	추출불가	미고려	수동
SRV	O	X	처리불가능	추출불가	미고려	수동
WHISK	O	X	처리가능	추출불가	미고려	수동
제안한 방법	O	X	처리가능	추출가능	고려	반자동

반의 엔티티 인식 방법을 사용하게 되면, 이전 단계까지는 인식이 되지 않았던 정보 소스의 아이템에 있어서 핵심적인 역할을 수행하는 타이틀을 효과적으로 인식하도록 할 수 있다.

표 2에서는 본 논문에서 제안한 시스템과 외국 시스템과의 기능 비교를 보인다. 8개의 시스템 모두 구조화된 문서를 대상으로한 정보추출 시스템이며, 자연어 문서에 대해서는 정보추출이 불가능하다. 다중슬롯은 시스템이 다중 슬롯을 갖는 정보를 추출할 수 있는지 여부를 나타내는 것이다. 즉, 여러 관련 정보를 통합할 수 있는지의 여부를 나타내는 것이다. 'WIEN', 'STALKER' 그리고 제안한 방법이 다중 슬롯을 처리가 가능하다. 표의 나머지 열인 레이블이 없는 경우, 하위링크고려, 지식확장 방법에 있어서 비교하면, 기존의 시스템보다 나은 기능을 보이고 있다.

5. 결론 및 향후 연구

본 논문에서는 인터넷에 존재하는 다양한 웹 정보 소스에서 효율적이고 정확하게 랩퍼를 생성할 수 있도록 하는 도메인 지식 기반의 확률적 랩퍼 생성 시스템을 제안하였다. 효율적이고 정확한 랩퍼 생성 시스템을 구축하기 위해서 반자동 도메인 지식의 확장을 도입했고, 테이블의 구조 정보를 이용했다. 또한 상세 정보로 연결되어 있는 하이퍼링크를 이용했고, 확률적 방법을 새롭게 모델링하여 레이블이 없어서 인식되지 않는 토큰의 엔티티를 식별하도록 하였다. 이렇게 여러 가지 방법을 적용함으로써, 사용자의 개입없이 다양한 정보 소스에 대해서 보다 추출 정확도가 높은 랩퍼를 생성할 수 있었다.

규칙 생성의 정확도 향상 관점에서 다루어질 수 있는 자동 도메인 지식의 확장은 웹의 빠른 변화와 동적인 특성을 반영하여 변화를 빠르게 수용할 수 있기 때문에, 새로운 포맷의 데이터들을 놓치지 않고 추출할 수 있는 장점이 있다. 또한 각 도메인별로 추출된 정보를 통합함으로써, 아이템에 대한 보다 상세한 정보를 얻게 하였다. 하나의 사이트에서 제공하는 정보의 양(속성 필드)이 제한되어 있다고 하더라도, 각각의 사이트에서 추출된 정보들을 통합하게 되면 서로의 부족한 부분을 채워주기 때문에 보다 상세하고 정확한 정보를 얻을 수 있게 된다. 이와 같은 일련의 작업들을 그래픽한 사용자 인터페이스를 이용하여 작업할 수 있도록 하였기 때문에 사용자가 편리하게 프로그램을 이용할 수 있다.

그러나 반자동 도메인 지식의 확장 부분에 있어서 레이블, 구분자, 값이 있을 경우에 값의 레이블이 도메인 지식의 어떤 엔티이에 포함될 것인가를 결정하는 부분에 있어서, 결정 규칙이 다소 불완전

한 상태이다. 학습 데이터를 이용하여 값이 가장 많이 존재하는 엔티티를 해당 값의 엔티티로 결정하도록 하고 있는데, 이 경우 값이 날짜 데이터일 경우에는 해당 값의 엔티티를 결정하기가 매우 힘들어 진다. 예를 들어, '시작일'과 '종료일'이 엔티티로 존재한다면 날짜 데이터를 어떤 엔티티로 식별할 것인가는 결정하기가 매우 어려워진다. 이와 같은 이유로 인해 반자동 도메인 지식의 확장은 다소 불완전하게 작동할 소지가 있다. 하지만 기존에 존재하지 않았던 새로운 포맷들을 감지하고 인식해 나가도록 하는 방법은 궁극적으로 시스템의 전체적인 커퍼리지를 높일 수 있게 될 것이다.

참 고 문 헌

[1] B. Adelberg, NoDoSE- A tool for Semi-Automatically Extracting Structured and Semistructured Data from Text Documents, ACM SIGMOD, 1998.

[2] A. Arasu, H. Garcia-Molina, Extracting structured data from web pages, ACM SIGMOD, 2003.

[3] R. Baumgartner, S. Flesca, G. Gottlob, Declarative Information Extraction, Web Crawling, and Recursive Wrapping with Lixto, Lecture Notes in Computer Science, 2001.

[4] A. Blum, T. Mitchell, Combining Labeled and Unlabeled Data with Co-Training, Proceedings of the 1998 Conference on Computational Learning Theory, 1998.

[5] D. Buttler, L. Liu, and C. Pu, A Fully Automated Object Extraction System for the World Wide Web, Proceedings of the 2001 International Conference on Distrubuted Computing Systems, May 2001.

[6] M. E. Califf. Relational Learning Techniques for Natural Language Information Extraction, PhD thesis, University of Texas at Austin, August 1998.

[7] F. Ciravegna. Learning to Tag for Information Extraction from Text, Workshop Machine Learning for Information Extraction, European Conference on Artifical Intelligence ECCAI, August 2000. Berlin, Germany, 2000.

[8] W. Cohen, M. Hurst, and L. S. Jensen. A flexible learning system for wrapping tables and lists in. html documents, The Eleventh International World Wide Web Conference WWW-2002, 2002.

[9] V. Crescenzi, G. Mecca, P. Merialdo, RoadRunner: Towards Automatic Data Extraction from Large Web Sites, Proceedings of 27th International Conference on Very Large Data Bases, 2001.

[10] L. Eikvil, Information Extraction from World Wide Web: A Survey, Report No. 945, ISBN 82-539-0429-0, July, 1999.

[11] D.W. Embley, D.M. Campbell, Y.S. Jiang, Y.-K. Ng, R.D. Smith, S.W. Liddle, D.W. Quass, A Conceptual-Modeling Approach to Extracting Data from the Web, International Conference on Conceptual Modeling / the Entity Relationship Approach, 1998.

[12] D. Freitag, Machine Learning for Information Extraction in Informal Domains, PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, November 1998.

[13] D. Freitag, N. Kushmerick. Boosted Wrapper Induction, Proceedings of the Seventh National Conference on Artificial, pages 577-583, 2000.

[14] J. R. Gruser, L. Raschid, M. E. Vidal, and L. Bright, Wrapper Generation for Web Accessible Data Sources, Proceedings of

the 3rd IFCIS International Conference on Cooperative Information Systems, New York, August, 1998.

[15] C. Hsu, and M. Dung, Generating Finite-State Transducers for Semi-Structured Data Extraction from the Web, Information Systems Vol. 23, No. 8, pp.521-538, 1998.

[16] C. N. Hsu, C. C. Chang, Finite-State Transducers for Semi-Structured Text Mining, Workshop on Text Mining IJCAI 99, 1999.

[17] M. Junker, M. Sintek, M. Rinck. Learning for Text Categorization and Information Extraction with ILP, Proc. Workshop on Learning Language in Logic, June 1999.

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

[19] N. Kushmerick, Wrapper induction: Efficiency and expressiveness, AAAI-98 Workshop on AI and Information Integration, July, 1998.

[20] N. Kushmerick, B. Thomas. Intelligent Information Agents R&D in Europe: An AgentLink perspective, chapter Adaptive Information Extraction: A Core Technology for Information Agents. Springer, 2002.

[21] L. Liu, C. Pu, and W. Han, XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources, Proceedings of the 16th International Conference on Data Engineering, 2000.

[22] Paolo Merialdo, Paolo Atzeni, Giansalvatore Mecca, Design and development of data-intensive web sites: The araneus approach, ACM Transaction on Internet Technology TOIT 3(1): 49-92, 2003.

[23] I. Muslea, S. Minton, and C. A. Knoblock, A Hierarchical Wrapper Induction for Semistructured Information Sources, Proceedings of the Third International Conference on Autonomous Agents, September 10, 1999.

[24] I. Muslea, Extraction patterns for information extraction tasks: a survey, Proceedings of AAAI'99: Workshop on Machine Learning for Information Extraction, 1999.

[25] B. A. Ribeiro-Neto, A. Laender, and A. Soares da Silva, Extracting semistructured data through examples, CIKM'99, 1999.

[26] S. Solderland, Learning Information Extraction Rules for Semi-structured and Free Text, http://www.cs.washington.edu/homes/solderland/WHISK.ps

[27] J. Yang, H. Seo, N. Koo, J. Choi, J. Kim, S. Kim, K. Lee, and H. Ham, A More Scalable Comparison Shopping Agent, Engineering of Intelligent Systems, pp.766-772, Paisely, Scotland, EIS 2000, 2000.

[28] 서희경, 양재영, 최중민, Semi-structured 문서의 Wrapper 자동 생성을 통한 정보통합에이전트, HCI2001 학술대회, pp. 794-799, 2001.

[29] 서희경, 양재영, 최중민, 준구조화 정보소스에 대한 지식기반 Wrapper 학습 에이전트, 정보과학회 논문지: 소프트웨어 및 응용, 29권, 1-2호, pp. 42-52, 2002.

[30] 최중민, 인터넷 정보추출 에이전트, 정보과학회지 18권 5호, pp. 48-53, 2000.

◐ 저 자 소 개 ◑

윤 보 현(Yun Bo Hyun)

1992년 목포대학교 전산통계학과 졸업(학사)
1995년 고려대학교 대학원 컴퓨터학과 졸업(석사)
1999년 고려대학교 대학원 컴퓨터학과 졸업(박사)
1999년~2003년 한국전자통신연구원 컴퓨터소프트웨어연구소 선임연구원 및 팀장
2003년~현재 목원대학교 컴퓨터교육과 교수
관심분야 : 자연어처리, 정보검색, 텍스트마이닝, 정보보호, 바이오인포매틱스, Natural Language Processing, Information Retrieval, Text Mining, Information, Security, Bioinformatics
E-mail : ybh@mokwon.ac.kr