도메인 지식 기반 랩퍼 생성의 추출 성능 향상에 관한 연구

Study on the Improvement of Extraction Performance for Domain Knowledge based Wrapper Generation

정 창 후* 최 윤 수** 서 정 현*** 윤 화 묵****
Chang-Hoo Jeong Yun-Soo Choi Jeong-Hyeon Seo Hwa-Mook Yoon

요 약

기존의 도메인 지식 기반의 랩퍼 학습 방법은 도메인에 대한 정보를 바탕으로 해당 정보 소스에 대한 랩퍼를 생성한다. 응용 분야에 맞게 정의된 도메인 지식을 이용함으로써 정보 소스에서 제공하는 다양한 텍스트의 의미와 형태를 이해할 수 있다. 그러나 정보 소스에서 제공되는 모든 텍스트에 의미 인식의 근거가 되는 레이블이 붙어서 제공되는 것이 아니기 때문에, 도메인 지식만을 이용해서 랩퍼를 학습하는 방법은 한계에 부딪힐 수밖에 없다. 이러한 문제를 해결하기 위해서 본 논문은 인터넷에 존재하는 다양한 웹 정보 소스에서 효율적이고 정확하게 랩퍼를 생성하는 도메인 지식 기반의 확률적 랩퍼 생성 시스템을 제안한다. 효율적이고 정확한 랩퍼 생성 시스템을 구축하기 위해서 도메인 지식뿐 아니라 상세 정보로 연결되어 있는 하이퍼링크와 엔티티 인식을 위한 확률 모델을 이용한다. 이와 같은 방법을 적용함으로써 사용자의 개입 없이 다양한 정보 소스에 대해서 보다 추출 성능이 좋은 랩퍼를 생성할 수 있다.

Abstract

Wrappers play an important role in extracting specified information from various sources. Wrapper rules by which information is extracted are often created from the domain-specific knowledge. Domain-specific knowledge helps recognizing the meaning the text representing various entities and values and detecting their formats. However, such domain knowledge becomes powerless when value-representing data are not labeled with appropriate textual descriptions or there is nothing but a hyper link when certain text labels or values are expected. In order to alleviate these problems, we propose a probabilistic method for recognizing the entity type, i.e. generating wrapper rules, when there is no label associated with value-representing text. In addition, we have devised a method for using the information reachable by following hyperlinks when textual data are not immediately available on the target web page. Our experimental work shows that the proposed methods help increasing precision of the resulting wrapper, particularly extracting the title information, the most important entity on a web page. The proposed methods can be useful in making a more efficient and correct information extraction system for various sources of information without user intervention.

☞ Keyword : 도메인 지식(Domain Knowledge), 랩퍼(Wrapper), 정보 추출(Information Extraction)

1. 서 론

최근 들어 인터넷의 급속한 성장과 보급으로 인해 사용자가 이용할 수 있는 정보의 양이 기하급수적으로 증가하였다. 숙련된 컴퓨터 기술 없이도 누구나가 인터넷에 웹사이트라는 것을 가질 수 있게 되었고, 많은 정보 서비스 업체들은 자신들의 고유 영역을 넓혀 가면서 다양한 컨텐츠 서비스들을 제시하고 있는 상황이다. 그러나 이와 같이 정보가 급증하면서 사용자가 얻을 수 있는 유용한 자원의 양이 많아진 것은 사실이지만 그만큼 실제적으로 원하는 핵심 데이터들을 찾기는 좀 더 어려워진 상황이다. 이러한 정보 과부하[1]는 사용자로 하여금 인터넷 이용에 대한 만족도를 떨어뜨릴 뿐만 아니라,

* 정회원 : 한국과학기술정보연구원 시스템개발팀
chjeong@kisti.re.kr
** 정회원 : 한국과학기술정보연구원 시스템개발팀
armian@kisti.re.kr
*** 정회원 : 한국과학기술정보연구원 시스템개발팀
jerry@kisti.re.kr
**** 정회원 : 한국과학기술정보연구원 시스템개발팀장
hmyoon@kisti.re.kr
[2006/02/14 투고 - 2006/03/16 심사 - 2006/07/20 심사완료]

정보 검색 기술로도 원하는 정보를 쉽게 찾을 수 없는 심각한 상황에 이르게 되었다. 이러한 인터넷의 질적 저하 상황에서 사용자가 원하는 정보만을 추출하여 제시하는 시스템에 대한 요구 사항이 크게 부각되고 있다.

인터넷에서의 정보 추출의 문제를 다루는 보편적인 접근법은 다양한 정보 소스에 접근하는 이질성을 캡슐화하는 랩퍼를 작성하는 것이다. 랩퍼(wrapper)는 특정한 정보 소스에 대해서 관심있는 데이터의 위치와 구조, 포맷 등을 나타내는 추출 규칙이라고 정의할 수 있다[2]. 기존의 도메인 지식 기반의 랩퍼 학습 방법은 도메인에 대한 정보를 바탕으로 해당 정보 소스에 대한 랩퍼를 생성한다. 도메인 지식(domain knowledge)은 특별한 응용 도메인을 위해서 폭넓게 사용되는 개념인 엔티티를 정의하고, 개념을 표현하기 위해서 사용되는 유사어의 집합인 레이블을 지정하며, 데이터들이 출현하는 형태인 포맷 등에 관련된 정보를 기술한다. 응용 분야에 맞게 정의된 도메인 지식을 이용함으로써 시스템은 정보 소스에서 제공하는 다양한 텍스트의 의미를 이해할 수 있고 구조 또한 자동으로 감지할 수 있다는 장점이 있다. 그러나 정보 소스에서 제공되는 모든 텍스트에 인식의 근거가 되는 레이블이 붙어서 제공되는 것이 아니기 때문에 도메인 지식만을 이용해서 랩퍼를 학습하는 방법은 한계에 부딪힐 수밖에 없다.

2. 관련 연구

랩퍼 생성(wrapper generation)에 관련된 연구로는 수동 랩퍼 생성과 반자동 랩퍼 생성, 그리고 자동 랩퍼 생성이 있다[3]. 랩퍼의 수동 생성은 추출해야 할 정보 소스의 추출 규칙을 사람이 일일이 기술하는 방법[4-6]을 의미하고, 랩퍼의 반자동 생성은 추출해야 할 정보 소스의 추출 규칙을 랩퍼의 디자인을 돕는 도구를 이용하여 반자동으로 생성하는 방법[7-9]을 의미한다. 그리고 랩퍼의 자동 생성은 주로 기계 학습 기술을 사용하여 프로그램이 자동으로 랩퍼를 생성하는 방법을 의미한다. 자동 랩퍼 생성을 위해서 다양한 학습 알고리즘들이 개발되어 왔는데, 이러한 알고리즘을 이용한 방법은 학습 단계를 거쳐야 하기 때문에 시간이 많이 소요될 수 있지만 랩퍼 생성 시에 발생하는 전문가의 노력을 최소화시킬 수 있다.

자동 랩퍼 생성의 연구로는 추출 가능한 정보 소스의 클래스를 구분해 놓고서 어떤 클래스에 속하는 지를 학습하는 방법[10]과 도메인 지식 기반의 학습 방법[11]이 있다. 전자는 자동으로 랩퍼를 생성하는 기술인 랩퍼 유도(wrapper induction)에 대해서 제안하는데, 빠르게 학습할 수 있는 여러 개의 랩퍼 클래스를 구분해 놓은 후에 정보 소스를 처리하도록 한다. 6개의 구분된 클래스가 있고 각각의 클래스 W에 대해서 랩퍼를 생성하는 알고리즘 learn-W를 제공한다. 후자는 적용 도메인별로 도메인 지식을 구축해 놓고 이것을 이용하여 각각의 정보 소스에 대한 랩퍼를 생성하는 방법에 대해서 제안하고 있다. 도메인 지식에 이미 추출되어야 할 정보들이 표현되어 있기 때문에 학습 데이터를 미리 구축해 놓을 필요가 없다. 하지만 이러한 도메인 지식을 이용하기 위해서는 정보 소스의 문서에 레이블이 있어야만 한다는 제약 조건이 있다.

본 논문에서는 인터넷에 존재하는 준구조화된 웹 정보 소스에서 효율적이고 정확하게 정보를 추출하는 도메인 지식 기반의 확률적 랩퍼 생성 시스템에 관해서 설명하도록 한다. 시스템의 추출 정확도를 높이기 위해 레이블이 없이 나오는 텍스트들에 대해서 해당 텍스트의 엔티티를 자동으로 인식할 수 있는 확률 모델을 제안한다. 이 방법은 도메인 지식 기반의 랩퍼 생성과 마찬가지로 인간의 개입을 최소로 요구하기 때문에 실세계의 응용에 보다 편리하게 적용시킬 수 있을 뿐만 아니라, 기존의 도메

인 지식만을 이용한 랩퍼 생성 시스템이 수행하지 못하는 단서가 없는 텍스트에 대해서도 엔티티 인식을 효과적으로 수행한다.

3. 랩퍼 생성

랩퍼 생성의 추출 성능 향상을 위해서 고려해 볼 수 있는 요소로는 다음과 같이 크게 두 가지가 있다.

3.1 하이퍼링크 활용 방법

많은 웹 정보 소스가 사용자에게 정보를 제공할 때, 처음에는 간략 정보만을 제공하는 방식을 취하고 있다. 그리고 나서 해당 아이템의 상세 정보 보기를 원했을 경우에만 하이퍼링크로 연결되어 있는 상세 정보를 보여주도록 한다. 이러한 방법은 사용자가 원하는 정보를 대략적으로 빨리 훑어볼 수 있게 해주는 장점이 있다. 또한 사용자가 처음에 접속하는 페이지에 많은 정보를 보여주기 위해서는 데이터베이스에서 한 번에 모든 정보를 가져와서 웹 페이지를 생성해야 하는데, 이럴 경우에 정보를 생성하도록 하는 웹 프로그램의 동작 시간이 길어질 수 있다. 이러한 방식은 정보 소스에 접근할 때 초기 접속 시간을 길어지게 하기 때문에, 사용자에게 서비스에 대한 불편을 초래할 수 있다. 따라서 아이템에 대한 충분한 정보를 얻기 위해서는 이러한 웹의 특성을 고려하여 하이퍼링크에 연결되어 있는 정보를 잘 활용하여야 한다.

하이퍼링크를 이용하기 위한 방법은 다음과 같다.

• 랩퍼 생성 시

- 메인 페이지에서 제공되는 정보들의 패턴을 분석하여 각 아이템의 바운더리를 감지한다.
- 감지된 바운더리 안에 있는 모든 하이퍼링크를 쫓아가서 유용한 정보가 있는 지를 확인한다. 도메인 지식을 이용해서 인식된 엔티티의 개수가 가장 많은 문서가 유용한 문서이다.
- 하이퍼링크에 연결되어 있는 문서의 정보가 유용하다고 판단되면 링크의 위치와 발견된 엔티티 관련 정보를 통합하여 랩퍼에 기록한다.

• 정보 추출 시

- 랩퍼를 읽어 들여 하이퍼링크에서 정보를 추출해야 하는 지를 결정한다.
- 메인 페이지에서 정보를 추출하고, 하이퍼링크의 정보 추출 표시가 있으면 하이퍼링크에 연결된 페이지에서도 정보를 추출한다.
- 메인 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.

3.2 확률 정보 활용 방법

정보 소스에서 랩퍼를 생성할 때, 레이블을 가지고 있는 텍스트는 도메인 지식에 의해서 자동으로 인식된다. 그러나 레이블을 가지고 있지 않는 텍스트는 도메인 지식을 이용하더라도 해당 텍스트에 대한 의미를 이해할 수 있는 단서가 없기 때문에, 텍스트에 대한 엔티티를 인식할 수가 없다. 이렇게 인식되지 않는 텍스트의 의미를 이해하기 위해서 확률적인 방법을 도입하도록 한다.

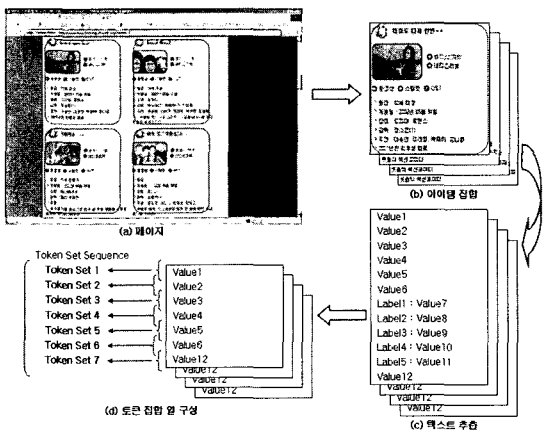
(1) 엔티티 인식 모델 배경

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 방법을 적용하여야 한다.

우선 모델을 제안하기 이전에 관련된 용어에 대해서 정의할 필요가 있다. 엔티티는 도메인에서 유용하게 사용될 수 있는 구성 요소의 기본 단위이다. 레이블은 해당 정보 소스에서 엔티티를 인식할 수 있도록 제공하는 단서이다. 아이템은 정보 소스에서 제공하는 정보의 기본 단위라고 정의할 수 있다. 대부분의 웹 정보 소스가 페이지에 여러 아이템을 리스트나 테이블과 같은 일정 패턴에 맞게 표시를 하고 있다. 아이템은 데이터베이스의 투플이라고도 정의할 수 있다. 웹 문서에 대한 구조 분석을 수행할 경우에 텍스트 조각들이 태그에 의해서 띄엄띄엄 떨어져서 나오게 되는데, 이러한 텍스트 조각들을 브라우저에서 보여지는 것과 같이 논리적으로 묶어서 의미를 가질 수 있는 텍스트로 재구성한다. 이렇게 구성된 텍스트에서 엔티티의 값이 될 수 있는 부분을 토큰이라고 부르도록 하겠다.

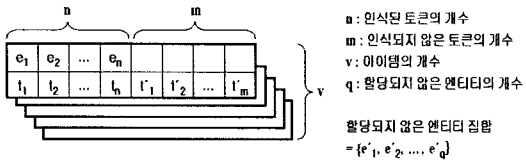
HTML 문서에 대한 구조 분석을 수행하면 많은 토큰들이 형성된다. 또한 이 과정에서 레이블이 있는 정보와 레이블이 없는 정보가 여러 아이템에 대해서 같은 패턴을 가지고 나오기 때문에, 정보 소스에 대해서 토큰 집합이라는 것을 구성할 수 있다. 즉, 하나의 아이템에 대해서 토큰을 하나 선택하면 다른 아이템의 같은 위치에 있는 토큰도 같은 역할을 하는 토큰으로 생각할 수 있기 때문에, 이러한 토큰들을 모아서 구성한 것을 토큰 집합(token set)이라고 말할 수 있다. 따라서 하나의 정보 소스에서 여러 개의 토큰 집합을 구성할 수 있다. 여러 개의 토큰 집합이 순차적으로 나오기 때문에 이것을 토큰 집합 열(token set sequence)이라고 부르도록 하겠다. 토큰 집합 열을 구성하는 과정을 살펴보면 (그림 1)과 같다.



〈그림 1〉 토큰 집합 열 구성 과정

(그림 1)의 (a)는 사용자가 HTML 문서를 브라우저에서 브라우징한 것으로서, 웹 정보 소스에서 흔히 볼 수 있는 형태이다. (b)는 브라우징된 정보 중에서 관심을 가지고 있는 정보만을 뽑아서 아이템 단위로 정보를 재배열한 것이다. 하나의 정보 소스에서 여러 개의 아이템이 나오는 것을 볼 수 있다. (c)는 웹 페이지의 구조를 분석하면서 일정한 패턴을 추출하여 하나의 아이템 단위로 텍스트 정보를 그룹핑한 것이다. 이러한 정보를 생성하기 위해서 문서에 대한 구조 분석을 수행할 때, (b)에서 보여지는 것과 같이 텍스트를 논리적으로 묶어 줘야 한다. 즉, 텍스트 사이에 존재하는 링크 태그나 폰트 태그와 같은 서식을 모두 제거하고 텍스트만을 추출해야 한다. (d)는 아이템에서 엔티티의 값이 될 수 있는 토큰들을 모아서 토큰 집합을 구성하는 것이다. 여러 개의 토큰 집합이 존재하기 때문에 토큰 집합 열을 구성할 수 있다는 것도 함께 보여준다.

(그림 1)의 아이템을 데이터베이스의 투플 개념으로 표현하면 (그림 2)와 같다.



〈그림 2〉 아이템의 투플 표현

(그림 2)에서 보여지는 것과 같이 아이템마다 m개의 인식되지 않은 토큰이 존재한다. 이렇게 인식되지 않은 토큰을 할당되지 않은 엔티티의 집합에 있는 각 엔티티로 어떻게 식별할 것인가가 모델에서의 핵심 요소라고 할 수 있다.

지금까지 설명한 것을 정리하면 다음과 같은 전제를 만들 수 있다.

① 하나의 아이템에 대해서 n개의 인식된 토큰이 있다. $\{t_1, t_2, \cdots, t_n\}$

② n개의 할당된 엔티티가 있다. $\{e_1, e_2, \cdots, e_n\}$

③ 하나의 아이템에 대해서 m개의 인식되지 않은 토큰이 있다. $\{t'_1, t'_2, \cdots, t'_m\}$

④ q개의 할당되지 않은 엔티티가 있다. $\{e'_1, e'_2, \cdots, e'_q\}$
이때 e'_k는 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 집합이 있다. 그리고 하나의 토큰 집합에는 v개의 토큰이 있다. $\{T_1, T_2, \ldots, T_n\}$, $T_i = \{t_{i1}, t_{i2}, \cdots, t_{iv}\}$

⑦ 하나의 정보 소스에 대해서 m개의 인식되지 않은 토큰 집합이 있다. 그리고 하나의 토큰 집합에는 v개의 토큰이 있다. $\{T'_1, T'_2, \ldots, T'_m\}$, $T_j = \{t'_{j1}, t'_{j2}, \cdots, t'_{jv}\}$

⑧ 도메인 지식에 (n + q)개의 엔티티를 원소로 갖는 집합 E가 정의되어 있다.

위에서 기술된 것을 바탕으로 토큰 집합에 엔티티 이름을 배타적으로 부여하는 확률 모델에 대해서 제안하고자 한다.

(2) 엔티티 인식 모델 설계

본 논문에서 제시하는 ERM(Entity Recognition Model)은 HMM(Hidden Markov Model)에서 아이디어를 얻은 것이다. HMM은 문장이 있을 때 문장의 구성 요소인 각 단어(word)에 품사(category)를 태깅하는 기능을 수행한다[12]. 본 논문에서 제안하는 ERM 역시 하나의 아이템을 구성하는 각 토큰(token)에 엔티티(entity)를 부여하는 기능을 수행한다. 다만 HMM과 다른 점은 HMM처럼 모든 단어에 확률적 방법을 적용하는 것이 아니라, 이미 레이블이 있어서 어떤 엔티티에 속하는지 결정이 된 토큰은 제외하고 그 외의 토큰에만 확률적 방법을 적용하도록 했다는 점이다. 또 다른 중요한 차이점은 HMM은 구성 요소의 순서, 즉 단어 간의 발생순서가 중요하게 고려되어야 하지만, ERM은 토큰 간의 발생순서가 중요하게 고려되지 않는다는 것이다. 따라서 비터비 알고리즘(Viterbi algorithm)과 같은 방법을 적용해서 계산할 확률에 대한 경우의 수를 줄여야 할 필요는 없다. 이러한 결정적 차이로 인해서 ERM의 수식은 HMM과는 다르게 구성된다. ERM과 HMM의 차이를 (표 1)과 같이 정리할 수 있다.

〈표 1〉 HMM과 ERM 비교

	HMM	ERM
Class	Category	Entity
Object	Word	Token
Probability	Word에 Category가 태깅될 확률	Token이 Entity로 식별될 확률
Corpus Statistics	Lexical Generation Probability	Model 1 Probability
Context Information	Bigram Probability	Model 2 Probability
Difference	Category의 발생순서가 중요함	Entity의 발생순서가 중요하지 않음

HMM

$= PROB(C_1,\cdots,C_T \mid w_1,\cdots,w_T)$

$\cong \prod_{i=1,T} PROB(w_i \mid C_i) * PROB(C_i \mid C_{i-1})$

ERM

$= PROB(e'_1,\cdots,e'_q \mid T'_1,\cdots,T'_m)$

$\cong \alpha * \{P(e'_i) * \frac{1}{v}\sum_{k=1}^{v} P(t'_{jk} \mid e'_i)\} +$

$(1-\alpha) * \{\frac{1}{v}\sum_{k=1}^{v}\sum_{h=1}^{n} P(e'_i = t'_{jk} \mid e_h = t_{hk}) * P(e_h = t_{hk})\}$

(단, 1 <= i <= q and 1 <= j <= m)

HMM에서 단어에 어떤 품사가 태깅될 확률을 나타내는 Lexical Generation Probability: $PROB(w_i \mid C_i)$는 ERM에서 베이지언 모델을 이용하는 Model 1 Probability: $P(e'_i) * \frac{1}{v}\sum_{k=1}^{v} P(t'_{jk} \mid e'_i)$와 같이 나타낼 수 있다. 베이지언 모델은 조건부 확률을 이용하는 방법으로서, 레이블이 없어서 인식되지 않는 토큰이 있을 때 토큰을 어떠한 엔티티로 식별하는 게 옳은 것인가를 결정하기 위해서 기존에 어떤 엔티티에서 어떤 토큰들이 식별되었는지를 역으로 관찰하는 방법이다. 단, 이때 웹페이지에서 출력되는 페이지에 여러 개의 아이템이 존재하기 때문에 하나의 토큰만을 고려하는 것이 아니라 각 아이템에 대해서 같은 위치에 나오는 모든 토큰들을 합쳐서 고려하도록 한다. 하나의 토큰이 어떤 엔티티로 식별되는 확률을 계산하는 것보다는 같은 성격을 가지고 있는 여러 개의 토큰이 어떤 엔티티로 식별되는 확률을 계산하는 것이 좀 더 변별력있는 확률을 구할 수 있기 때문이다. 이러한 개념을 이용하면 정보 소스의 아이템에 대해서 레이블이 없어서 식별되지 않는 토큰들을 확률 값을 이용하여 새로운 엔티티로 할당할 수 있다.

또한 HMM에서 두 개의 Category가 연속적으로 나타날 확률을 나타내는 Bigram Probability: $PROB(C_i \mid C_{i-1})$는 ERM에서 컨텍스트 정보를 이용하는 Model 2 Probability: $\frac{1}{v}\sum_{k=1}^{v}\sum_{h=1}^{n} P(e'_i = t'_{jk} \mid e_h = t_{hk}) * P(e_h = t_{hk})$와 같이 나타낼 수 있다. 하나의 아이템 안에 같이 존재하는 주변 정보들을 이용하는 방법으로서, 레이블이 없어서 인식되지 않은 토큰이 있을 때 토큰을 어떠한 엔티티로 식별하는 게 옳은 것인가를 결정하기 위해서 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 적용이 가능하다. 즉, 여러 정보소스에 대해서 랩퍼를 생성하고 정보를 추출하도록 하고 있기 때문에 다른 정보소스에서 추출된 정보를 이용하여 현재 정보 소스에서 문제가 되고 있는 것들을 해결할 수 있다.

결과적으로 $PROB(e'_1,\ldots,e'_q \mid T'_1,\ldots,T'_m)$가 가장 큰 확률 값을 갖는 e'_i 를 선택하여 토큰 집합 T'_j 의 엔티티로 할당한다. 단, 이때 토큰이 엔티티가 될 확률이 임계값(threshold)을 넘지 않을 경우에는 해당 토큰의 엔티티 식별은 무효로 한다. 임계값에 의해서 정보 소스에서 실제로 중요하게 사용될 수 있는 토큰인지 별로 의미가 없는 토큰인지를 구별해 내도록 한다. 임계값은 실험에 의해서 추정하도록 했다.

끝으로 처음의 토큰 집합 열로부터 토큰 집합 T'_j 를 제거하여 새로운 토큰 집합 열 T'_1, T'_2, ..., T'_{m-1} 을 생성한다. 새롭게 생성된 토큰 집합 열에 대해서 위의 과정을 반복해서 적용한다. 과정 중에 발생할 수 있는 차이는 인식되지 않은 엔티티 중의 하나가 새롭게 할당되어서 더 이상 할당이 불가능하기 때문에, 나머지 e'_i 에 대한 $P(e'_i)$ 값이 갱신될 필요가 있다는 것이다.

참고로 ERM은 HMM에서와 같이 각 구성 요

소간의 순서에 대한 제약 사항이 없으므로, 두 개의 확률을 가중치 변수 α를 이용하여 결합하도록 했다. 베이지언 모델을 이용하는 Model 1과 컨텍스트 정보를 이용하는 Model 2가 나름대로의 타당성있는 가치를 지니기는 하지만, 각각 상대적인 가중치를 두어 두 가지 모델을 혼합함으로써 좀 더 신뢰성있는 그리고 여러 가지 정보가 혼합된 견고한 모델을 구성할 수 있다. Model 1과 Model 2의 상대적인 중요도를 나타내는 α값은 실험을 통하여 적당한 값으로 추정하도록 한다.

4. 실 험

본 논문에서 제안한 알고리즘을 영화 도메인에 속하는 7개의 정보 소스(Site A, Site B, ..., Site G)에 적용시켜 보았다. 영화에 관련된 도메인 지식을 구축할 때 시스템의 응용 분야에 맞게 도메인 지식의 엔티티를 적절히 선택해야 한다. 그러나 본 논문에서는 엔티티 인식 모델에 대한 평가를 목적으로 하기 때문에, 영화에 관련된 최대 도메인 지식을 가지고 실험을 수행하였다. 본 논문에서 정의한 영화 도메인의 엔티티는 제목, 장르, 감독, 출연, 등급, 제작, 각본, 촬영, 음악, 상영시간, 시작일 그리고 종료일로 구성되어 있다. (그림 1)의 (a)는 제목, 등급, 시작일(개봉일), 장르, 감독, 출연(주연)의 엔티티를 제공하는 정보 소스의 한 예이다.

4.1 실험 방법

본 논문에서 제안한 몇 가지 방법들의 유용성을 검증하기 위해서 실험을 단계적으로 수행하였다. 즉, 처음에는 도메인 지식만을 적용하여 랩퍼를 생성하도록 하였고, 다음에는 하이퍼링크에 대한 처리를 추가하여 랩퍼를 생성하도록 하였다. 마지막으로 본 논문에서 가장 중요하게 생각하는 인식되지 않는 토큰들에 대한 엔티티 인식 알고리즘을 적용하여 랩퍼를 생성하여 그 결과를 비교하였다.

각 사이트의 추출 성능의 정확도는 다음과 같이 계산된다.

정확도(precision) = (추출된 엔티티의 개수/추출해야 될 엔티티의 개수)*100

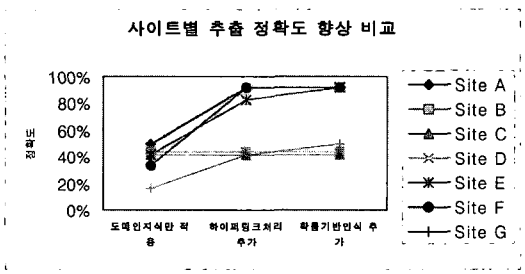
여기서 추출된 엔티티의 개수는 랩퍼를 학습하면서 인식된 엔티티의 개수라고 볼 수 있고, 추출해야 될 엔티티의 개수는 해당 도메인에서 정의한 엔티티의 개수라고 볼 수 있다. 참고로 본 실험의 영화 도메인 지식에서 정의된 최대 12개의 엔티티를 모두 제공하는 정보 소스는 없었기 때문에 추출 정확도가 100%인 경우는 존재하지 않았다.

전체 사이트에 대한 평균 정확도는 다음과 같이 계산된다.

평균 정확도(average precision) = (각 사이트의 정확도/평가를 수행한 사이트의 수)

4.2 결과 및 분석

사이트별 추출 성능 향상에 대한 결과는 (그림 3)와 같다.

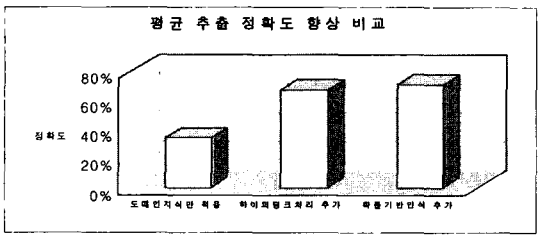


〈그림 3〉 사이트별 추출 정확도 향상 비교

각 사이트에서 새로운 방법을 추가함에 따라 추출 정확도가 점점 향상되는 것을 확인할 수 있다. 그러나 모든 사이트에서 각 단계의 추출

정확도가 항상 증가하는 것은 아니다. 연결된 하이퍼링크가 없는 경우에는 하이퍼링크에 대한 처리를 수행하여도 새로운 엔티티를 찾을 수 없고, 또한 모든 엔티티에 레이블이 존재하는 경우에는 인식되지 않는 토큰들에 대한 엔티티 인식 알고리즘을 적용하여도 새롭게 엔티티를 식별해 낼 수가 없다. Site B와 Site C가 전형적으로 이 경우에 해당한다. 이와 같은 사이트는 도메인 지식만을 적용하여도 충분히 처리할 수 있는 경우이다.

전체적인 추출 성능 향상에 대한 결과는 (그림 4)과 같다.



〈그림 4〉 평균 추출 정확도 향상 비교

첫 번째 실험에서는 도메인 지식만을 적용하여 랩퍼를 생성하도록 하였다. 실험 결과 해당정보 소스에서 추출할 수 있는 엔티티들에 대해서 적절하게 랩퍼를 생성하는 것을 관찰할 수 있었다. 그러나 이러한 방법은 웹사이트가 가지고 있는 하이퍼링크의 유용성을 제대로 활용하지 못한 결과를 초래하였다. 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 적절했다는 것을 보여준다.

여기서 새롭게 인식된 엔티티의 성격을 살펴볼 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, 굳이 레이블을 붙이지 않는 이유도 있다고 보여진다. 이와 같은 상황에서 확률 기반의 엔티티 인식 방법을 사용하면 이전 단계까지는 인식이 되지 않았던 정보 소스의 아이템에 있어서 핵심적인 역할을 수행하는 타이틀을 효과적으로 인식하는 것을 살펴볼 수 있다.

또한 실험을 통하여 부가적으로 얻어진 중요한 사실은 정보 소스에 따라 Model 1과 Model 2의 크기가 서로 다르다는 것이다. 실험에서 확인해 본 결과, 어떤 사이트에서는 Model 1의 확률 값이 크게 나왔고, 또 다른 사이트에서는 Model 2의 확률 값이 크게 나왔다. 이것은 사

이트마다 나오는 정보의 특성 때문으로 생각된다. 베이지언 모델을 이용하는 Model 1보다 컨텍스트 정보를 이용하는 Model 2의 확률 값이 크게 나오는 정보 소스의 경우, 데이터 안에 레이블이 있는 텍스트가 상대적으로 많이 나오는 것을 확인할 수 있었다. 이러한 현상은 레이블이 많은 데이터가 컨텍스트 정보도 많이 가지고 있다는 것을 의미한다. 결국, Model 1과 Model 2의 상대적인 중요성은 정보 소스에서 나오는 데이터의 특성에 의존한다고 볼 수 있다. 따라서 가중치 변수 α값은 적용하려고 하는 정보 소스의 데이터 특성에 맞게 어느 Model에 비중을 둘 것인지를 적절히 고려하여 선택하도록 한다.

5. 결론 및 향후 연구

결과적으로 도메인 지식을 이용하여 랩퍼를 생성하는 시스템은 그 나름대로 많은 장점을 가지고 있음에도 불구하고 레이블이 없는 텍스트 인식에 있어서는 치명적인 약점을 가지고 있기 때문에, 확률적인 방법을 적용한 랩퍼 생성시스템은 그 중요성이 아주 크다고 볼 수 있다. 더군다나, 확률적인 방법을 적용해서 새롭게 인식하고 있는 텍스트의 대부분이 해당 아이템의 식별자가 될 수 있는 타이틀 역할의 엔티티가 많다는 점은 본 연구에서 제시한 방법론이 아주 유용하고 효과적이었다는 것을 입증하고 있다.

향후 연구로는 정보 소스별로 추출된 개별적 결과의 통합에 관련된 작업과 규칙 생성의 정확도 향상 관점에서 다루어질 수 있는 도메인 지식의 자동 확장에 관련된 작업이 이루어져야 할 것이다. 서로 다른 정보 소스에서 추출된 정보들은 각기 서로 다른 아이템을 표현하는 경우가 많지만 동일한 아이템에 대해서 서로 다른 엔티티들을 가지고 있는 경우도 있기 때문에, 추출된 아이템의 식별자를 기반으로 데이터를 통합해서 정제된 정보를 생성할 필요가 있다. 또한 도메인 지식이 초기에 제대로 구축이 되었다고 하더라도 웹과 같은 정보 소스의 빠른 변화와 동적인 특성으로 인해 해당 도메인에 대한 새로운 특성들이 나타날 수 있기 때문에, 정보를 추출하는 과정에서 발견되는 해당 도메인의 새로운 특성들을 자동으로 감지하여 도메인 지식을 확장할 수 있는 방법의 개발이 필요하다.

참 고 문 헌

[1] P. Maes, "Agents that Reduce Work and Information Overload", Communication of the ACM, vol.37, no. 7, pp. 31-40, July 1994.

[2] M. Hearst, "Information Integration", IEEE Intelligent Systems, vol.13, no.5, pp. 12-24, 1998.

[3] L. Eikvil, "Information Extraction from World Wide Web", A Survey, 1999

[4] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo, "Extracting Semistructured Information from the Web", Proceedings of Workshop on Management of Semistructured Data, pp. 18-25, 1997.

[5] J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. Breunig, and V. Vassalos, "Template-based Wrappers in the TSIMMIS System", ACM SIGMOD International Conference on Management of Data, pp. 532-535, 1997.

[6] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J.Widom, "The TSIMMIS Project: Integration of Heterogeneous Information Sources", Proceedings of Tenth Anniversary Meeting of the Information Processing Society of Japan, Tokyo, Japan, 1994.

[7] 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

[8] N. Ashish, and C. Knoblock, "Wrapper Generation for Semi-structured Internet Sources", Workshop on Management of Semistructured Data, Ventana Canyon Resort, Tucson, Arizona.

[9] N. Ashish, and C. Knoblock, "Semi-automatic Wrapper Generation for Internet Information Sources", Proceedings of Coopis Conference, 1997.

[10] N. Kushmerick, D. Weld, and R. Doorenbos, "Wrapper induction for information extraction", International Joint Conference on Artificial Intelligence (IJCAI), Nagoya, Japan, 1997.

[11] H. Seo, J. Yang, and J. Choi, "Knowledge-based Wrapper Generation by Using XML", IJCAI-2001 Workshop on Adaptive Text Extraction and Mining (ATEM 2001), pp. 1-8, Seattle, USA, 2001.

[12] James Allen, "Natural Language Understanding (2nd Edition)", pp. 189-204, Addison-Wesley Publishing Co, 1995

◐ 저 자 소 개 ◑

정 창 후 (Chang-Hoo Jeong)
1999년 충남대학교 컴퓨터과학과 졸업(학사)
2002년 충남대학교 대학원 컴퓨터과학과 졸업(석사)
2003년~현재 한국과학기술정보연구원 시스템개발팀
관심분야 : 정보검색 및 추출, 디지털 도서관, 메타데이터 레지스트리
E-mail : chjeong@kisti.re.kr

최 윤 수 (Yun-Soo Choi)
1993년 충남대학교 컴퓨터공학과 졸업(학사)
1995년 충남대학교 대학원 컴퓨터공학과 졸업(석사)
1995년~현재 한국과학기술정보연구원 선임연구원
관심분야 : 데이터베이스, 정보검색
E-mail : armian@kisti.re.kr

서 정 현 (Jeong-Hyeon Seo)
1987년 한양대학교 수학과 졸업(학사)
2003년 연세대학교 대학원 정보통신학과 재학(박사)
2000년~현재 한국과학기술정보연구원 선임연구원
관심분야 : 정보검색, 자연어처리
E-mail : jerry@kisti.re.kr

윤 화 묵 (Hwa-Mook Yoon)
1992년 서울산업대학교 전자계산학과 졸업(학사)
1997년 공주대학교 대학원 전자계산학과 졸업(석사)
2005년~현재 배재대학교 재학(박사)
현재 한국과학기술정보연구원 시스템개발팀장
관심분야 : 데이터베이스, 정보검색, 온톨로지
E-mail : hmyoon@kisti.re.kr