확률 기반 웹 콘텐츠 마이닝
Probabilistic based Web Contents Mining

윤보현*, 조광문**
목원대학교*, 목포대학교**

Yun Bo-Hyun*, Cho Kwang-Moon**
Mokwon University*, Mokpo National University**

요약

웹문서에 대한 콘텐츠 마이닝에서 레이블이 없는 엔티티 인식과 하위정보 및 추출결과의 정보통합은 중요하다. 본 논문에서는 레이블이 없는 엔티티를 인식하기 위해 베이지언 모델에 기반한 확률 기반 인식 방법을 제안한다. 또한 웹문서에 존재하는 하위링크정보를 이용하고, 추출한 중복된 결과를 통합할 수 있는 방안을 제시한다. 실험결과, 확률기반 엔티티인식과 정보통합을 수행한 방법이 가장 우수한 성능을 보임을 알 수 있다.

Abstract

In Web contents mining, it is important to recognize the unlabeled entities and to integrate the sub-linked information and the extracted results. This paper presents the probabilistic based method which can recognize the unlabeled entity by using the Baysien model. Moreover, we propose the method that can use the information of the sub-linked web pages and integrate the extracted results. In the experimental results, we can see that the probabilistic based entity and information integration show the most significant precision.

I. 서 론

웹 콘텐츠 마이닝의 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[5].

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.

웹은 링크정보를 가지고 있어서 콘텐츠 마이닝에 있어서 이러한 링크정보를 고려해야만 한다. 아울러 마이닝의 결과가 중복될 수 있기 때문에 정보통합 과정이 필요하다. 본 논문에서는 이러한 정보통합을 수행하는 콘텐츠 마이닝 방법을 제안하고자 한다.

II. 관련연구

자동 랩퍼 생성 방법은 크게 기계학습 방법[3-5], 데이터 마이닝 방법[1], 그리고 개념 모델링 방법[2]으로 나뉜다.

기계학습 방법은 인터넷상의 많은 정보들이 상관관계가 있는 데이터로 존재하고 있다고 보고, 라벨이 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 인스턴스)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

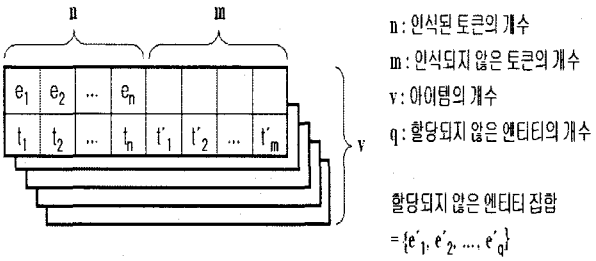
다르지만 도메인 지식과 일치하는 엔티티만을 인식하고, 레이블이 없거나 다른 이름의 레이블이 존재하면 엔티티를 인식하지 못한다. 아울러 하위링크나 추출결과에 대한 정보통합이 이루어지지 않는다.

Ⅲ. 베이지언 모델 기반 엔티티 인식

3.1 모델 용어 정의

우선 모델을 제안하기 이전에 관련된 용어에 대해서 정의할 필요가 있다. "엔티티"는 도메인에서 유용하게 사용될 수 있는 구성 요소의 기본 단위이다. 예를 들어 제목이나 감독, 혹은 주연과 같은 정보들이 엔티티가 될 수 있다. 레이블은 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 부르도록 하겠다. 구조 분석을 수행하면 많은 토큰들이 생기게 된다.

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, 다른 아이템의 같은 위치에 있는 토큰도 같은 역할을 하는 토큰으로 생각할 수 있기 때문에(각각의 토큰들은 해당 아이템의 동일 엔티티를 설명하는 텍스트가 될 것이다), 이러한 토큰들을 모아서 구성한 것을 "토큰 집합(Token Set)"이라고 말할 수 있다. 따라서 하나의 정보 소스에서 여러 개의 토큰 집합을 구성할 수 있게 된다. 여러 개의 토큰 집합이 순차적으로 나오기 때문에 이것을 "토큰 집합 열(Token Set Sequence)"이라고 부르도록 하겠다. 아이템을 데이터이스의 투플 개념으로 표현하면 그림 1과 같다.



▶▶ 그림 1. 투플 구성

그림 1에서 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. $\{t_1, t_2, ..., t_n\}$
2. n개의 할당된 엔티티가 있다. $\{e, e_2, ..., e_n\}$
3. 하나의 아이템에 대해서 m개의 인식되지 않은 토큰이 있다. $\{t'_1, t'_2, ..., t'_m\}$
4. q개의 할당되지 않은 엔티티가 있다. $\{e'_1, e'_2, ..., e'_m\}$

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. $\{T_1, T_2, ..., T_n\}\ T_i = \{t_{i1}, t_{i2}, ...t_{iv}\}$
7. 하나의 정보 소스에 대해서 m개의 인식되지 않은 토큰 집합이 있다. 그리고 하나의 토큰 집합에는 v개의 토큰이 있다. $\{T'_1, T'_2, ..., T'_m\}\ T_j = \{t'_{j1}, t'_{j2}, ..., t'_{jv}\}$
8. 도메인 지식에 (n + m)개의 엔티티가 있다.

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

3.2 엔티티 인식

베이지언 모델은 조건부 확률을 이용하는 방법으로서, 레이

블이 없어서 인식되지 않는 토큰이 있을 때 토큰을 어떠한 엔티티로 식별하는 게 옳은 것인가를 결정하기 위해서, 기존에 어떤 엔티티에서 어떤 토큰들이 나왔나를 역으로 관찰하는 방법이다. 단, 이때 하나의 토큰만을 고려하는 것이 아니라, 여러 개의 아이템이 존재하기 때문에, 각 아이템에 대해서 같은 위치에 나오는 모든 토큰들을 합쳐서(토큰 집합을 구성해서) 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 있게 된다.

다음과 같은 과정을 거쳐서 모델의 확률 값을 계산할 수 있다.

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

$$P(e'_i|t'_j)=\frac{P(t'_j|e'_i)*P(e'_i)}{P(t'_j)} \equiv P(e'_i)*P(t'_j|e'_i) \quad \text{--- ①}$$

$$P(e'_i|T'_j) \equiv P(e'_i)*P(T'_j|e'_i) \equiv P(e'_i)*\frac{1}{v}\sum_{k=1}^{v}P(t'_{jk}|e'_i) \quad \text{--- ②}$$

토큰이 엔티티에 속할 확률 값은 ①과 같이 계산한다. 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 ②와 같이 계산하도록 한다.

3.1) $P(t'_{jk}|e'_i)$는 학습 데이터로부터 얻어진다. $P(t'_{jk}|e'_i)$= 엔티티 e'_i의 값이 t'_{jk}인 개수 / 엔티티 e'_i 가 나온 개수

3.2) $P(e'_i)$는 학습 데이터로부터 얻어진다. $P(e'_i)$ = 할당되지 않은 엔티티 e'_i가 나온 개수 / 할당되지 않은 엔티티가 나온 모든 개수

4) $P(e'_i|T'_j)$가 가장 큰 확률 값을 갖는 e'_i를 선택하여, 토큰 집합 T'_j의 엔티티로 할당한다. 단, 이때 토큰이 엔티티가 될 확률이 임계 값(Threshold)을 넘지 않을 경우에는 해당 토큰의 엔티티 식별은 무효로 한다. 임계 값에 의해서 정보 소스에서 실제로 중요하게 사용될 수 있는 토큰인지, 별로 의미가 없는 토큰인지를 구별해 내도록 한다. 임계 값은 실험에 의해서 추정하도록 했다.

5) 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 단계 3과 4를 반복해서 적용한다. 과정 중에 발생할 수 있는 차이는, 인식되지 않은 엔티티 중의 하나가 새롭게 할당되어 져서 더 이상 할당이 불가능하기 때문에, 나머지 e'_i에 대한 $P(e'_i)$값이 갱신되어질 필요가 있다는 것이다. 즉, 이미 할당된 e'_i 외의 나머지 엔티티에 대해서 $P(e'_i)$값은 더 커지게 된다.

모델을 적용시켜서 확률 값을 계산하는 방법은 다음과 같다.

1) 학습 데이터를 구축한다.

e_1 - e_n은 도메인 지식의 엔티티의 이름을 나타낸다. t_1 - t_m은 지금까지 발견되었던 토큰을 나타낸다. x_{11} - x_{mn}은 해당 토큰이 해당 엔티티에서 발생한 빈도 수를 나타낸다. s_{c1} - s_{cn}은 해당 엔티티에서 토큰이 발생한 전체 빈도 수를 나타낸다. s_{r1} - s_{rm}은 해당 토큰이 각각의 엔티티에서 발생한 전체 빈도 수를 나타낸다. s_{rc}는 토큰이 엔티티에서 발생한 빈도 수의 총합을 의미한다.

2) 확률 값을 계산한다.

정보 소스에서 인식되지 않은 토큰 t'_1이 엔티티 e'_2에 속할 확률 값은 다음과 같이 계산된다. 이때 e'_4에서 e'_n까지는 이미 인식이 된 엔티티라고 가정한다.

$$P(e'_2|t'_1)=\frac{P(t'_1|e'_2)*P(e'_2)}{P(t'_1)} \equiv P(t'_1|e'_2)*P(e'_2)=\frac{x_{12}}{s_{c2}}*\frac{s_{c2}}{s_{c1}+s_{c2}+s_{c3}}$$

Ⅳ. 정보통합

4.1 하위링크 정보통합

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 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.
- 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

4.2 추출 결과 정보 통합

웹 사이트 하나만을 가지고 볼 때, 해당 웹 사이트에서 구조 분석을 수행하고, 수행한 결과를 이용하여 랩퍼를 생성하고, 생성된 랩퍼를 이용하여 정보를 추출하고 추출된 정보를 저장하는 작업은 바람직한 과정이라고 볼 수 있다. 그러나 웹 사이트가 여러 개가 존재할 경우에는 각각의 웹 사이트에서 추출하여 가져온 결과가 중복되는 아이템을 생성할 수 있기 때문에 추출 결과를 바로 저장하는 것은 문제가 될 수 있다. 단순히 영화 도메인만을 보더라도, 여러 곳의 예매 사이트에서 동일한 극장의 동일한 영화를 예매 대상으로 할 수 있기 때문에 항상 다른 사이트의 추출 결과와 비교를 수행해야만 한다. 따라서 중복되는 아이템을 제거하는 비교 작업이 후처리로 반드시 포함되어야만 한다.

도메인 지식에서 정의된 통합 키를 이용하여 사이트 별로 추출된 서로 다른 정보들을 통합 템플릿을 생성하여 통합한다.

• 추출 결과 통합을 위한 처리 과정
- 도메인 지식에 통합을 위한 기본 키 정보를 표시한다.
- 추출 결과들의 기본 키 정보를 비교한다. 기본 키가 여러 개일 경우에는 여러 개의 키를 함께 비교하도록 한다.
- 기본 키가 같다고 판명된 아이템들에 대해서 필드 통합을 시도한다.
- 통합된 최종 결과 리스트를 생성한다.

V. 실험 및 결과

본 논문에서는 영화 도메인에 속한 7개의 웹 정보 소스에 대해서 배치 파일을 구성하였다. 이렇게 구성된 배치 파일의 정보 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.

[표 1] 실험에 사용된 웹 사이트

도메인	사이트 URL
Site A	http://www.corecine.co.kr
Site B	http://www.joycine.com
Site C	http://www.maxmovie.com
Site D	http://www.cinewel.com/
Site E	http://www.nkino.com
Site F	http://www.ticketpark.com
Site G	http://www.yesticket.co.kr

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.

각 사이트의 추출 성능의 정확도는 다음과 같이 계산된다.

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

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.

표 2에서는 7개의 사이트에 대한 정확도의 실험결과이다. Baseline 방법은 도메인 지식에 존재하는 레이블과 정확히 매치하는 엔티티만을 추출하는 방법이다. 정보통합은 하위정보통합과 추출결과 정보통합을 의미하고, 확률기반 인식방법은 정보통합방법과 확률기반 엔티티 인식 방법을 혼합한 방법이다. 이 방법이 평균 94%의 정확도로 가장 우수한 성능을 보이고 있다.

[표 2] 정확도 실험결과

	Baseline	정보통합	확률 기반 인식
Site A	53%	89%	95%
Site B	46%	93%	94%
Site C	54%	90%	93%
Site D	65%	88%	92%
Site E	51%	93%	95%
Site F	48%	92%	94%
Site G	55%	89%	95%

표 3에서는 본 논문에서 제안한 시스템과 외국 시스템과의 기능 비교를 보인다. 8개의 시스템 모두 구조화된 문서를 대상으로한 정보추출 시스템이지만 제안한 방법이 레이블이 없는 엔티티를 인식할 수 있으며, 하위링크 및 추출결과를 통합 할 수 있음을 보이고 있다.

[표 3] 다른 시스템과의 비교

	레이블이 없는 엔티티	하위링크 정보통합	추출결과 정보통합
ShopBot	추출불가	X	X
WIEN	추출불가	X	X
SoftMealy	추출불가	X	X
STALKER	추출불가	X	X
RAPIER	추출불가	X	X
SRV	추출불가	X	X
WHISK	추출불가	X	X
제안한 방법	추출가능	O	O

Ⅵ. 결 론

▌참 고 문 헌▐

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

[2] 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.

[3] 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.

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

[5] 윤보현, 구조화된 웹 문서에 대한 자동 정보추출, 한국인터넷정보학회논문지, 제6권, 제3호, pp.129-145, 2005.