웹페이지에서 레이블이 없는 텍스트 인식을 위한 확률 모델

정창후* 이민호* 주원균* 맹성현**
한국과학기술정보연구원*, 한국정보통신대학원대학교**
{chjeong, cokeman, joo}@kisti.re.kr,
myaeng@icu.ac.kr

A Probabilistic Method for Recognizing Unlabeled Text on Web Pages

Chang-Hoo Jeong*, Min-Ho Lee*, Won-Kyun Joo*, Sung-Hyon Myaeng**
Korea Institute of Science and Technology Information*,
Information and Communications University**

요 약

도메인 지식은 텍스트의 포맷과 의미 정보를 이용하여 웹에 존재하는 텍스트의 다양한 의미를 이해할 수 있도록 도와준다. 그러나 도메인 지식은 텍스트에 데이터의 의미를 표현하는 레이블이 존재하지 않을 경우에 텍스트 인식을 제대로 수행할 수 없기 때문에 무용지물이 되고 만다. 이러한 문제를 해결하기 위해 본 논문에서는 레이블이 존재하지 않는 텍스트의 의미를 효과적으로 추론할 수 있는 엔티티 인식 모델을 제안한다. 엔티티 인식 모델은 베이지언 모델과 컨텍스트 정보를 결합한 방법으로서, 구조 분석을 수행한 HTML 문서의 텍스트 토큰에 대해서 어떤 엔티티에 속할 것인가를 결정하는 기능을 수행한다. 실험 결과 본 모델을 사용할 경우 기존에는 레이블이 없어서 인식되지 않았던 텍스트들을 효과적으로 인식하는 것을 확인할 수 있었다.

1. 서 론

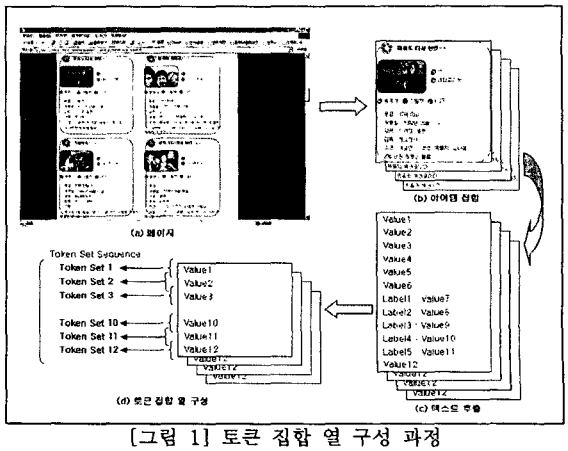
정보 소스에서 텍스트에 대한 의미를 이해할 때, 레이블을 가지고 있는 텍스트는 도메인 지식에 정의되어 있는 포맷 정보와 의미 정보를 이용하여 자동으로 인식되게 된다[1]. 그러나 레이블을 가지고 있지 않는 텍스트는 도메인 지식을 이용한다고 하더라도, 해당 텍스트에 대한 의미를 이해할 수 있는 단서가 없기 때문에, 텍스트에 대한 엔티티를 인식할 수가 없게 된다. 본 논문에서는 이렇게 인식되지 않는 텍스트의 의미를 이해하기 위해서 확률적인 방법을 새롭게 모델링하고자 한다.

2. 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 일정 패턴(리스트 형태나 테이블 형태)에 맞게 표시를 하고 있다. 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 브라우저에서 보여지는 것과 같이 논리적으로 묶어서 의미를 가질 수 있는 텍스트로 재구성하게 된다. 이렇게 구성된 텍스트에서 엔티티의 값이 될 수 있는 부분을 토큰이라고 부르도록 하겠다.

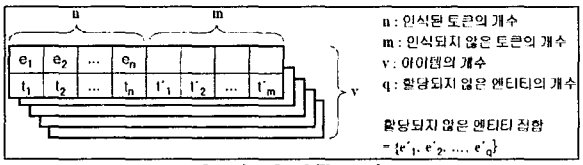
HTML 문서에 대한 구조 분석을 수행하면 많은 토큰들이 생기게 된다. 또한 이 과정에서 레이블이 있는 정보와 레이블이 없는 정보가 모든 아이템에 대해서 같은 패턴을 가지고 나오기 때문에, 정보 소스에 대해서 토큰 집합이라는 것을 구성할 수 있다. 즉, 하나의 아이템에 대해서 토큰을 하나 선택하면, 다른 아이템의 같은 위치에 있는 토큰도 같은 역할을 하는 토큰으로 생각할 수 있기 때문에(각각의 토큰들은 해당 아이템의 동일 엔티티를 설명하는 텍스트가 될 것이다), 이러한 토큰들을 모아서 구성한 것을 토큰 집합(Token Set)이라고 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. 토큰 집합 열을 구성하는 과정을 살펴보면 [그림 1]과 같다.



[그림 1] 토큰 집합 열 구성 과정

[그림 1]의 (a)는 HTML 문서를 브라우져에서 브라우징한 것으로서, 사용자가 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] 투플 구성

[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. $\{t_1, t_2, ..., t_n\}$
2. n개의 할당된 엔티티가 있다. $\{e_1, e_2, ..., e_n\}$
3. 하나의 아이템에 대해서 m개의 인식되지 않은 토큰이 있다. $\{t'_1, t'_2, ..., t'_m\}$
4. q개의 할당되지 않은 엔티티가 있다. $\{e'_1, e'_2, ..., e'_q\}$ 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. $\{T_1, T_2, ..., T_n\}$, $T_i = \{t_{i1}, t_{i2}, ..., t_{iv}\}$
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 + q)개의 엔티티가 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 부착하는 기능을 수행한다[2]. 본 논문에서 제안하는 ERM 역시 하나의 아이템을 구성하는 각 토큰(Token)의 엔티티(Entity)를 식별하는 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 단어간의 발생 순서가 중요하게 고려되어야 하지만, ERM은 토큰간의 발생 순서가 중요하게 고려되지 않는다. 따라서 Viterbi Algorithm과 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.

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

[표 1] HMM과 ERM 비교

$$HMM$$
$$= PROB(C_1,...,C_T \mid w_1,...,w_T)$$
$$\cong \prod_{i=1,T} PROB(w_i \mid C_i) * PROB(C_i \mid C_{i-1})$$

$$ERM$$
$$= PROB(e'_1,...,e'_q \mid T'_1,...,T'_m)$$
$$\cong \alpha * \{P(e'_i) * \frac{1}{v}\sum_{k=1}^{v} P(t'_{jk} \mid e'_i)\} +$$
$$(1-\alpha) * \{\frac{1}{v}\sum_{k=1}^{v}\sum_{h=1}^{n} P(e'_i = t'_{jk} \mid e_h = t_{hk}) * P(e_h = t_{hk})\}$$

(단, 1 <= i <= q and 1 <= j <= m)

HMM에서 하나의 Word가 Category가 될 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. 계산하는 방법은 다음과 같다.

1. $P(t'_{jk} \mid e'_i)$는 통계 데이터로부터 얻어진다. $P(t'_{jk} \mid e'_i)$ = 엔티티 e'_i 의 값이 t'_{jk} 인 개수 / 엔티티 e'_i 가 나온 개수
2. $P(e'_i)$는 통계 데이터로부터 얻어진다. $P(e'_i)$ = 할당되지 않은 엔티티 e'_i가 나온 개수 / 할당되지 않은 엔티티가 나온 모든 개수

또한 HMM에서 두 개의 Category가 연속적으로 나타날

확률을 나타내는 Bigram Probability: $PROB(C_i|C_{i-1})$는 ERM에서 컨텍스트 정보를 이용한 Model 2 Probability: $\frac{1}{v}\sum_{k=1}^{v}\sum_{h=1}^{n}P(e'_i=t'_{jk}|e_h=t_{hk})*P(e_h=t_{hk})$ 와 같이 나타낼 수 있다. 계산하는 방법은 다음과 같다.

3. $P(e'_i=t'_{jk} \mid e_h=t_{hk})$는 통계 데이터로부터 얻어진다. $P(e'_i=t'_{jk} \mid e_h=t_{hk})$ = 엔티티 e'_i 값이 t'_{jk} 이고, 엔티티 e_h 값이 t_{hk} 인 투플의 개수 / 통계 데이터의 전체 투풀 개수

4. $P(e_h=t_{hk})$는 통계 데이터로부터 얻어진다. $P(e_h=t_{hk})$ = 엔티티 e_h 값이 t_{hk} 인 투플의 개수 / 통계 데이터의 전체 투풀 개수

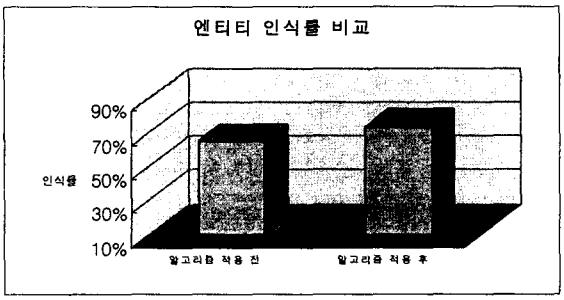
5. $PROB(e'_1,...,e'_q \mid T'_1,...,T'_m)$가 가장 큰 확률 값을 갖는 e'_i 를 선택하여, 토큰 집합 T'_j 의 엔티티로 할당한다. 단, 이 때 토큰이 엔티티가 될 확률이 임계 값을 넘지 않을 경우에는 해당 토큰의 엔티티 식별은 무효로 한다. 임계 값에 의해서 정보 소스에서 실제로 중요하게 사용될 수 있는 토큰인지, 별로 의미가 없는 토큰인지를 구별해 내도록 한다. 임계 값은 실험에 의해서 추정하도록 했다.

6. 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 1, 2, 3, 4, 5를 반복해서 적용한다. 과정 중에 발생할 수 있는 차이는, 인식되지 않은 엔티티 중의 하나가 새롭게 할당되어 져서 더 이상 할당이 불가능하기 때문에, 나머지 e'_i 에 대한 $P(e'_i)$ 값이 갱신되어 질 필요가 있다는 것이다. 즉, 이미 할당된 e'_i 외의 나머지 엔티티에 대해서 $P(e'_i)$ 값은 더 커지게 된다.

참고로 ERM은 HMM에서와 같이 각 구성 요소간의 순서에 대한 제약 사항이 없으므로, 두 개의 확률을 가중치 변수 α를 이용하여 결합하도록 했다. 실제적인 α값의 결정은 적용하려는 도메인의 데이터 특성에 맞게 적절히 조절하도록 한다.

4. 엔티티 인식 모델 실험

본 논문에서 제안한 알고리즘을 영화 도메인의 정보 소스 7개의 사이트(Site A, Site B, ..., Site G)에 적용시켜 보았다. 영화에 관련된 도메인 지식을 구축할 때 시스템의 응용 분야에 맞게 도메인 지식의 엔티티를 적절히 선택하도록 해야 한다. 그러나 본 논문에서는 엔티티 인식 모델에 대한 평가를 목적으로 하기 때문에, 영화 도메인에서 생각해 볼 수 있는 모든 엔티티를 포함하도록 하여, 영화에 관련된 최대 도메인 지식을 가지고 실험을 수행하도록 했다.



[그림 3] 엔티티 인식 비율 비교

사이트 구분	레이블 개수	Model 1	확률 비교	Model 2
Site D	2	0.0007380	>	0.0004231
Site E	5	0.0004548	<	0.0010652
Site G	2	0.0014716	>	0.0006424

[그림 4] Model 1과 Model 2의 확률 비교

실험 결과 [그림 3]과 같이 정보 소스에서 인식할 수 있는 텍스트의 수가 증가하는 것을 관찰할 수 있었다. 영화 도메인에서는 대부분의 텍스트에 레이블을 붙여 주는 경우가 많기 때문에 증가율이 크지는 않았지만, 레이블이 없이 나오는 텍스트들에 대해서는 엔티티 인식이 제대로 수행됨을 확인할 수 있었다. 또한 [그림 4]와 같이 정보 소스에 따라 확률 Model 1과 2의 값이 상대적인 차이가 발생하는 것을 볼 수 있었다. 영화 도메인에 속하는 7개의 사이트에 대해서 실험해 본 결과 어떤 사이트에서는 Model 1의 확률 값이 크게 나오는 것을 볼 수 있었고, 또 다른 사이트에서는 Model 2의 확률 값이 크게 나오는 것을 볼 수 있었다. 이것은 사이트마다 나오는 정보의 특성 때문으로 생각되어 진다. 실제로 Site E의 데이터를 분석해 본 결과, 레이블이 있는 텍스트가 다른 사이트에 비해 많이 나오는 것을 확인할 수 있었다([그림 4]에서 보여지는 것처럼 Site E에는 레이블이 있는 텍스트가 5개가 나오는 것에 반해, Site D와 Site G는 레이블이 있는 텍스트가 2개 밖에 안 나옴). 다시 말해서 보통의 경우에는 Model 1의 확률 값이 Model 2의 확률 값보다 크게 나오는데, 특별히 레이블이 있는 텍스트가 많아서 컨텍스트 정보를 많이 갖게 되는 경우에는 Model 2의 확률 값이 크게 나오는 것을 관찰할 수 있었다. 따라서 정보 소스의 데이터 특성에 의존하여 각 Model의 확률 값이 다르게 나온다는 것을 알 수 있었다. 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 고려하여 적절히 선택하도록 한다.

5. 결론 및 향후 연구

본 논문에서 제안하는 확률 모델을 영화 도메인에 적용하여 실험을 수행한 결과 인식할 수 있는 엔티티의 수가 증가하는 것을 확인할 수 있었다. 이것은 레이블이 없는 토큰들에 대해서 확률적 방법을 적용해서 엔티티 인식을 수행한 방법이 효과적이었음을 보여준다. 향후 연구로는 본 모델을 보다 다양한 도메인에 적용시켜 일반성을 갖출 수 있도록 하는 작업이 필요하다고 보여진다.

6. 참고 문헌

[1] 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.

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