Journal of Korean Institute of Intelligent Systems, Vol. 24, No. 3, June 2014, pp. 285-291
http://dx.doi.org/10.5391/JKIIS.2014.24.3.285

테이블로부터 링크드 데이터 생성을 위한 패턴 충돌 해소

Conflict Resolution of Patterns for Generating Linked Data From Tables

한용진* · 김권양** · 박세영*†
Yong-Jin Han, Kweon Yang Kim, and Se Young Park†

*경북대학교 컴퓨터학부, **경일대학교 컴퓨터공학과
† School of Computer Science and Engineering, Kyungpook National University

요 약

최근 링크드 오픈 데이터(예, RDF, OWL)를 이용해 대량의 테이블로부터 새로운 링크드 데이터를 생성하기 위한 연구가 주목을 받고 있다. 본 논문은 이러한 링크드 데이터 생성을 위해 패턴을 이용한 방법을 제안한다. 패턴을 이용한 방법은 근본적으로 패턴들 간의 충돌 문제를 안고 있다. 예를 들어, 어떤 테이블 헤더(header)를 서로 다른 링크드 데이터 속성들로 맵핑하는 패턴들은 서로 충돌한다. 기존의 연구들은 통계적으로 우세한 패턴을 적용하여 정확도의 감소를 감수하거나 정확도를 높이기 위해 충돌하는 패턴들을 무시해 왔다. 제안하는 방법은 주어진 테이블에 적용되는 패턴들을 연계함으로써 모든 헤더들에 대한 적합한 패턴들을 찾는다. DBPedia와 위키피디아의 테이블을 이용한 실험에서 제안한 방법이 패턴 충돌을 효과적으로 해소하는 결과를 보였다.

키워드 : 링크드 데이터, 충돌 해소, 패턴, 테이블

Abstract

Recently, many researchers have paid attention to the study on generation of new linked data from tables by using linked open data (e.g. RDF, OWL). This paper proposes a new method for such generation of linked data. A pattern-based method intrinsically has a conflict problem among patterns. For instance, several patterns, mapping a single header of a table into different properties of linked data, conflict with each others. Existing studies have sacrificed precision by applying a statistically dominant pattern or have ignored conflicting patterns to increase precision. The proposed method finds appropriate patterns for all headers in a given table by connecting patterns applied to the headers. Experiments using DBPedia and Wikipedia showed results that conflicts of patterns are effectively resolved by the proposed method.

Key Words : Linked Data, Conflict Resolution, Pattern, Table

1. 서 론

테이블은 문서나 웹 페이지 안에서 중요한 정보를 간결하고 이해하기 쉽게 표현한다. 예를 들어, 어떤 사람의 이름, 나이, 출생일 등의 정보를 문장으로 표현하면 하나의 장문 혹은 여러 개의 문장으로 작성해야 한다. 반면, 이름, 나이, 출생일 등의 속성을 테이블의 헤더(header)로 하고 대응하는 속성 값을 헤더와 이웃하게 둠으로써 간결하고 이해하기 쉽게 표현할 수 있다. Google의 연구[1]에 따르면 이러한 관계적 정보를 표현한 테이블들이 1억 5천만 개 이상 웹 상에 존재한다. 또한 정부 기관들에서 날씨, 재해 등의 정보를 테이블 형태로 제공하는 것이 일반적이다 [2].

이러한 테이블들은 균일한 스키마로 표현되지 않기 때문에 검색을 하거나 정보를 통합하여 활용하는데 한계가 있다 [3]. 예를 들어, 객체의 속성을 표현하는 헤더는 다양한 어휘로 표현될 수 있다. 따라서, 동일한 의미를 가지는 서로 다른 표현의 헤더를 식별하지 못한다면, 대량의 테이블 데이터를 효과적으로 활용하기 어렵다.

최근, 이러한 문제를 해결하기 위한 방법으로 테이블 데이터로부터 새로운 링크드 데이터를 생성하는 연구들이 활발히 진행되고 있다 [3, 4, 5]. 이들 연구는 테이블의 헤더를 링크드 데이터의 속성(property)으로, 테이블의 값을 링크드 데이터의 객체 혹은 값으로 맵핑하여 테이블의 의도된 의미를 명시적으로 표현한다. 따라서, 서로 다른 출처의 다

접수일자: 2014년 1월 28일
심사(수정)일자: 2014년 3월 28일
게재확정일자 : 2014년 3월 28일
† Corresponding author
본 연구는 미래창조과학부 및 한국산업기술평가관리원의 SW컴퓨팅산업원천기술개발사업(SW)의 일환으로 수행하였음. [10044494, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발]

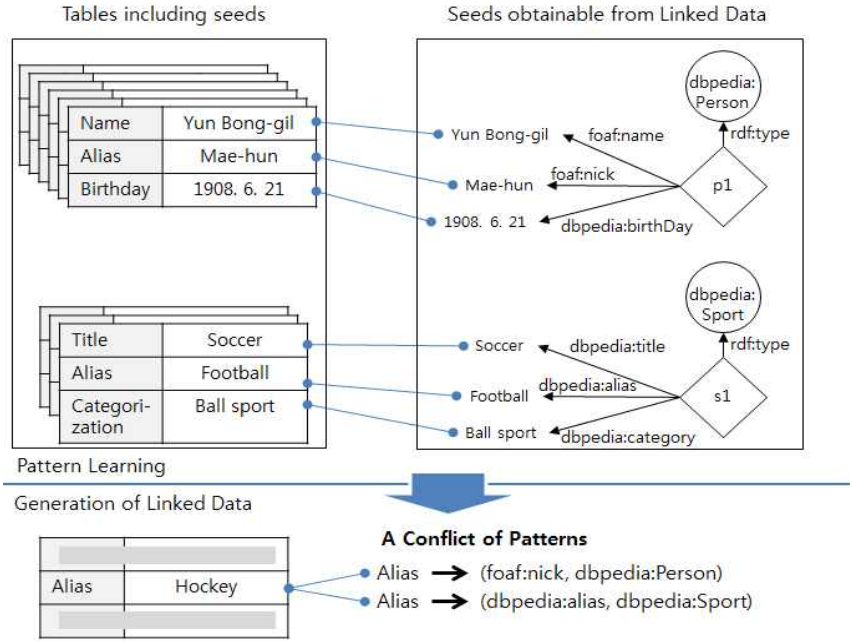

그림 1. 패턴 기반 방법의 패턴 충돌 문제

Fig. 1. A conflict problem of patterns in a pattern-based method

른 스키마로 동일한 유형의 정보를 제공하는 테이블들을 링크드 데이터의 속성을 이용해 균일하게 표현할 수 있다.

본 논문은 이러한 연구들 중에서 이미 알려진 링크드 데이터를 활용한 패턴 기반 방법에 주목하고 있다[5, 6]. 패턴 기반 방법은 주어진 관계를 가지는 객체들을 입력으로 하여 이들이 등장하는 테이블들로부터 공통적인 패턴을 찾는다. 다음으로, 찾은 패턴을 이용하여 새로운 테이블로부터 해당 관계를 가지는 새로운 객체들을 추출한다. 이러한 패턴 기반 방법은 초기에 주어진 객체들을 seed로 하여 패턴 학습과 정보 추출을 반복하면서 대량의 객체를 효과적으로 추출할 수 있다. 하지만, 패턴 기반 방법은 패턴들이 충돌하는 문제를 본질적으로 안고 있다.

그림 1은 테이블로부터 링크드 데이터 생성을 위한 패턴 기반 방법의 패턴 충돌 문제를 보인 것이다. 그림1의 상단은 seed 객체를 이용한 패턴 학습을 묘사한 것이다. Yun Bong-gil(윤봉길)은 개념, dbpedia:Person (사람)에 속하는 객체로서 이름(foaf:name), 별칭(foaf:nick), 생일(dbpedia:birthDay)에 대한 정보를 가지고 있다. 패턴 학습을 위해 이러한 정보를 포함하는 테이블들을 찾는다. Seed가 등장하는 여러 개의 테이블들을 얻고 이러한 테이블들의 공통점을 분석하여 사람의 속성에 대응하는 테이블의 헤더를 찾을 수 있다. 예를 들어, 인물관련 테이블로부터 Alias라는 헤더와 foaf:nick이 공통적으로 매칭되는 것을 관찰하여, Alias가 dbpedia:Person 개념에 대한 속성 foaf:nick에 대응하는 패턴을 찾게 된다. 한편, 개념, dbpedia:Sport의 객체들을 seed로 하여 Alias가 속성 dbpedia:alias에 맵핑되는 패턴도 찾을 수 있다. 이들 두 패턴은 동일한 헤더 표현에 대해 서로 다른 속성으로의 맵핑을 정의하고 있다. 따라서, 그림 1의 하단과 같이 어떤 새로운 테이블이 주어졌을 때, Alias에 대한 맵핑 속성들이 충돌하게 된다.

이러한 패턴 충돌이 발생 했을 때, 기존의 방법들은 크게 두 가지 형태로 동작한다. 하나는 통계적으로 지배적인 패턴을 적용하는 것이다. 예를 들어, 그림 1에서 인물 관련 테이블이 다량 존재하므로 Alias를 foaf:nick으로 맵핑하는 패턴을 선택하고 대응하는 값 Hockey를 foaf:nick의 값으로 할당한다. 이러한 접근은 항상 Alias를 foaf:nick으로 맵핑하기 때문에 스포츠관련 테이블에서의 정보 추출 정확도를 떨어뜨리게 된다. 다른 하나는 충돌하는 패턴들을 무시하는 것이다. 즉, 패턴 충돌 문제를 피함으로써 객체 정보 추출의 정확도를 높이게 된다. 하지만 이러한 경우, 추출할 정보에 대한 커버리지가 떨어진다.

제안하는 방법은 주어진 테이블에 함께 등장하는 헤더들에 대한 맵핑 패턴을 고려함으로써 패턴 충돌을 해소한다. 예를 들어, 그림 1에서 Alias 주변에 “Title”과 “Categorization”이라는 헤더가 있다면 해당 테이블은 스포츠에 관한 것이고 Alias를 dbpedia:alias로 맵핑하는 패턴을 선택한다. 선택된 패턴을 이용해 객체 정보를 추출하고, 인스턴스 및 속성 관계를 생성한다. 이러한 접근의 기본적인 가정은 특정 테이블은 동일한 유형의 객체를 표현한다는 것이다. DBPedia와 위키피디아의 테이블을 이용한 실험에서 제안한 방법을 통해 패턴 충돌을 효과적으로 해소하는 결과를 보였다.

2. 관련 연구

실시간으로 쏟아져 나오는 방대한 량의 웹 문서는 링크드 데이터를 확장하기 위한 유용한 자원으로 활용되고 있다. 많은 연구들은 웹 문서를 bag-of-words로 간주하여 정보 추출 관점에서 링크드 데이터의 개념 혹은 관계에 대한 객체 정보 생성 방법을 제안하였다 [7, 8, 9]. 이러한 접근은 웹 문서의 많은 부분을 차지하는 비구조적인 텍스트를 통계적으로 처리할 수 있는 장점이 있지만, 웹 문서 내의 테이블과 같은 구조적인 정보가 손실되는 한계가 있다. 테이블

의 구조 정보를 활용하기 위해서는 웹 문서로부터 유의미한 테이블을 찾고 [10], 동일한 유형의 서로 다른 테이블들을 통일된 형태로 통합하는 과정이 요구 된다 [11]. 본 논문은 유의미한 테이블들이 주어졌을 때, 이들 테이블들로부터 통일된 형태의 링크드 데이터를 생성하는 방법에 초점을 맞추고 있다.

기존 연구들은 기본적으로 테이블의 헤더는 링크드 데이터의 속성 혹은 개념으로, 값은 속성의 값 혹은 객체로 맵핑함으로써 링크드 데이터를 확장한다. 최근에는 이미 알려진 링크드 데이터를 이용한 방법들이 활발히 연구되고 있다, 이러한 연구들은 링크드 데이터를 활용하는 방법에 따라 크게 두 가지 접근으로 나눌 수 있다. 하나는 상향식(bottom-up) 접근으로 하나의 테이블이 주어졌을 때, 테이블의 의미를 파악하기 위해 이미 알려진 링크드 데이터를 활용하는 것이다. 최근 소개된 그래피컬 모델 기반의 방법들은 특정 테이블에 등장하는 정보들의 연관성을 분석하는데 초점을 두고 있다 [3, 5].

다른 하나는 하향식(top-down) 접근으로 추출하고자 하는 객체의 개념 혹은 이들의 관계를 입력으로 대량의 테이블들로부터 해당 개념 혹은 관계에 대한 객체들을 추출하는 것이다. 이러한 접근은 텍스트를 대상으로 한 정보 추출을 위해 활발히 연구되어 왔다. 예를 들어, Espresso[12]는 어떤 관계에 해당하는 객체 쌍들을 이용해 텍스트로부터 해당 관계를 표현하는 패턴들을 찾고, 찾은 패턴들을 이용해 새로운 객체 쌍들을 찾는다. 이러한 과정을 반복함으로써 코퍼스로부터 주어진 관계에 해당하는 다량의 객체들을 추출한다. PROSPERA [13]는 이러한 bootstrapping 접근에 링크드 데이터의 추론 기술을 적용하여 텍스트를 대상으로 한 최신의 정보 추출 성능을 보였다 [13].

본 논문은 이러한 bootstrapping 접근을 반구조 정보에 적용한 연구에 주목한다. SEAL [6]의 경우, HTML 소스로부터 자동차, 사람 등에 해당하는 객체들을 추출하기 위한 패턴을 찾는다. 이를 위해 seed가 등장하는 주변의 문자열로부터 패턴을 추출하고 서열화하여 적합한 패턴을 찾는다. 이러한 bootstrapping 접근은 패턴 추출과 객체 추출이 반복될수록 패턴의 의미가 전이(semantic drift)되는 문제를 안고 있다. 의미 전이는 객체 추출의 정확도를 떨어뜨리는 원인이 된다.

이러한 의미 전이를 줄이기 위한 최신 연구로서 CSEAL [4]이 소개되었다. CSEAL은 특정 테이블의 정보가 맵핑되는 링크드 데이터의 속성들의 연결 관계를 고려함으로써 객체 추출의 정확도를 높인다. 예를 들어, 어떤 패턴들을 적용한 결과로, 특정 테이블 내의 헤더들이 서로 직접적으로 연결되지 않는 속성들에 대응할 때, 해당 패턴들을 적용하지 않는다.

제안하는 방법은 패턴 충돌을 해소한다는 측면에서 기존 연구들과 차별화된다. SEAL의 경우, 기본적으로 단일한 관계 혹은 속성에 대한 정보 추출 방법이다. 따라서, 여러 관계 혹은 속성 정보를 추출하는 경우, 추출된 인스턴스들 간의 모호성이 발생할 수 있다. 즉, 특정 테이블 값이 서로 다른 속성들의 값으로 추출될 수 있다. 한편, CSEAL은 충돌하는 패턴들을 제거하기 때문에 제안하는 방법과 차이가 있다. 제안하는 방법과 CSEAL의 차이는 다음 장에서 좀 더 구체적으로 설명한다.

3. 충돌 해소 기반 링크드 데이터 생성

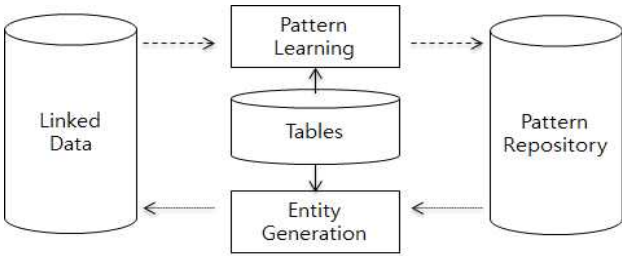

그림 2. 패턴을 이용해 테이블로부터 링크드 데이터 생성
Fig. 2. Generation of Linked Data from Tables using patterns

그림 2는 테이블로부터 링크드 데이터를 생성하기 위한 bootstrapping 접근을 묘사한 것이다. Bootstrapping 접근은 패턴 학습 단계와 객체 생성 단계를 반복하는 것이다. 패턴 학습 단계에서는 링크드 데이터의 객체들을 seed로 하여 주어진 테이블들로부터 패턴을 추출하고, 추출된 패턴들을 서열화하여 가정 적합한 패턴들을 저장소에 저장한다. 객체 생성 단계에서는 학습된 패턴들을 이용하여 대량의 테이블들로부터 객체를 생성한다. 생성된 객체들은 링크드 데이터의 인스턴스로 저장된다. 이러한 접근은 CSEAL과 동일한 프레임워크에 해당한다. 하지만 패턴의 형태 및 실제 인스턴스를 추출하는 알고리즘에는 차이가 있다.

그림 3과 4는 CSEAL-like 링크드 데이터 생성 알고리즘과 제안한 방법의 알고리즘을 각각 보인 것이다. CSEAL-like 알고리즘은 원래 CSEAL 알고리즘을 본 논문에서 사용하는 패턴의 형태에 적합하게 수정한 것이다. 이 알고리즘은 링크드 데이터의 각 속성에 대해 패턴 생성(줄: 7)과 객체 생성(줄: 8)을 순차적으로 수행한다. 이후 학습된 패턴들은 이전에 학습된 패턴들과 충돌이 발생하는지를 확인하고, 충돌이 발생한 경우 해당 패턴을 제거한다. 이때, 주어진 패턴에 의해 추출된 객체들이 기존 패턴들에 의해 추출된 인스턴스들과 상호 배타적인(mutually exclusive) 관계를 가질 때, 패턴이 충돌하는 것으로 간주한다. 예를 들어, 책에 대한 속성으로서 '저자'와 회사에 대한 속성으로서 'CEO'는 모두 사람을 값으로 가지지만, 서로 다른 개념들에 대한 속성이기 때문에 상호 배타적인 관계를 가진다. 이러한 접근은 충돌하는 패턴들을 제거함으로써 객체 추출의 정확도를 높일 수 있지만, 특정 헤더에 대한 대량의 정보들이 무시될 수 있다.

제안하는 방법은 패턴 학습과 객체 생성을 모든 속성들을 대상으로 적용한다. 즉, 그림 4에서 모든 속성들에 대한 패턴 학습을 수행(줄: 5-9)한 후, 각 테이블에 대해 객체 생성을 수행(줄:10-21)한다. 객체 생성 시, 객체 후보들을 추출된 테이블에 따라 하나의 그룹으로 묶고(줄: 11), 해당 그룹 내에서 상호 공존하는 관계를 고려하여 패턴 충돌을 해소(줄: 15)한다. 즉, 적용된 패턴들의 개념 유형(type)에 따라 추출 후보들을 각각의 그룹으로 묶고, 각 그룹에 대한 점수를 매겨 후보들을 선택한다. 제안하는 방법의 기본적인 가정은 주어진 테이블의 헤더들은 동일한 개념에 대한 속성들에 대응한다는 것이다. 이러한 가정에 따라 그림 4의 알고리즘은 테이블의 헤더들을 특정 개념의 속성들로 맵핑한다. 따라서, CSEAL에서 무시하는 패턴들을 활용한다는 측

```
1: Input: A linked data L, and a set of tables T
2: Output: Instances / patterns for each predicate
3: for i = 1 to ∞ do
4:     for each predicate p ∈ L do
5:         QUERY tables containing recently promoted
6:         instances
7:         LEARN patterns for each table returned
8:         EXTRACT new candidates using patterns
9:         FILTER patterns that conflict with previously
10:        learned patterns
11:        RANK candidate instances
12:        PROMOTE top candidates
13:    end for
14: end for
```

그림 3. CSEAL-like 링크드 데이터 확장 알고리즘
Fig. 3. A CSEAL-like algorithm for generating linked data

면에서 링크드 데이터 확장의 재현율을 높일 수 있다. 또한, 특정 테이블 내 헤더들의 관계 정보를 고려함으로써 SEAL에서 문제가 되는 의미 전이를 줄일 수 있다. 이것은 결과적으로 링크드 데이터 생성의 정확율을 높이게 된다.

3.1 패턴 학습

링크드 데이터의 하나의 객체 x가 있을 때, x의 개념을 c, x의 특정 속성을 p와 p의 값을 v라고 하자. 본 논문에서 패턴 학습을 위한 seed를 (c, p, v)로 표현한다.

어떤 seed, s=(c, p, v)가 주어졌을 때, 패턴 학습을 위해 속성값 v가 등장하는 테이블을 찾는다. 패턴은 v에 해당하는 테이블 값의 헤더 h를 결합하여 트리플 (h, c, p)로 표현한다. 따라서 어떤 개념 c에 속성 p가 정의되었을 때, p는 개념 c의 모든 하위 개념들과 결합하여 별개의 패턴으로 정의된다. 이렇게 속성을 가질 수 있는 개념을 명시함으로써 추출될 객체의 개념을 세부적으로 한정할 수 있다.

패턴 생성은 전체 테이블 데이터를 입력으로 하여 각 개념 및 각 속성에 대해 독립적으로 수행된다. 따라서 어떤 테이블의 헤더에 대해 둘 이상의 트리플 패턴이 생성될 수 있다. 이러한 경우는 크게 두 가지로 나누어 볼 수 있다. 하나는 동일한 헤더를 가지는 두 개의 패턴이 같은 개념을 가지면서 서로 다른 속성을 표현하는 경우이다. 예를 들어, 저자와 번역가를 표현하는 속성 Writer와 Translator에 대해 각각 (Person, Writer, 이외수), (Person, Translator, 이외수)라는 seed가 주어졌다고 하자. '저자'라는 헤더의 값으로 '이외수'를 포함하는 테이블로부터 두 개의 패턴들 (저자, Writer, Person)와 (저자, Translator, Person)이 생성된다. 이러한 경우, seed로부터 발현되는 빈도수가 가장 높은 패턴을 선택한다.

다른 하나는 동일한 헤더를 가지는 두 개의 패턴이 서로 다른 개념을 가지는 경우이다. 예를 들어, 서론에서 소개하였던 그림 1의 Alias는 dbpedia:Person와 dbpedia:Sport을 각각 개념으로 하는 두 개의 패턴에 포함된다. 이들 패턴들은 모두 패턴 저장소에 저장한다. 이후 객체 생성 단계에서 충돌 해소를 통해 적합한 패턴을 선택하여 적용한다.

모든 선택된 패턴들은 발현되는 통계에 따라 점수를 부여하고 패턴 저장소에 저장한다. 트리플 패턴 t=(h, c, p)가 주어졌을 때, s에 대한 점수는 다음의 식 1과 같이 계산한다.

```
1: Input: A linked data L, and a set of tables T
2: Output: Instances / patterns for each predicate
3: for i = 1 to ∞ do
4:     T' ← φ
5:     for each predicate p ∈ L do
6:         QUERY tables containing recently promoted
7:         instances
8:         LEARN patterns for each document returned
9:     end for
10:    EXTRACT new candidates using patterns
11:    SORT&GROUP the candidates according to tables
12:    for each table t ∈ T do
13:        G ←candidates obtained from t
14:        if exist a conflict in G then
15:            GROUP&SCORE candidates in G according
16:            to type
17:            PROMOTE selected candidates
18:        else
19:            PROMOTE all candidates in G
20:        end if
21:    end for
22: end for
```

그림 4. 충돌 해소 기반 링크드 데이터 생성 알고리즘
Fig. 4. A conflict resolution based algorithm for generating linked data

$$Score(t) = \frac{n((h,c,p))}{n((h,c,*))} \quad (1)$$

n(·)는 입력된 패턴의 개수를 반환한다.

그림 5은 패턴 학습 예를 보인 것이다. 패턴 학습은 여러 개의 seed와 전체 테이블을 입력으로 하여 seed가 발현되는 테이블을 찾고 트리플 패턴을 생성한다. 패턴 선택 및 점수 계산을 효과적으로 하기 위해 관계형 데이터 베이스 형태로 패턴을 저장하고 빈도수를 계산한다. 최종적으로 선택된 패턴은 점수와 함께 패턴 저장소에 저장된다.

3.2 객체 생성

헤더 h, 개념 c, 속성 p로 구성된 트리플 패턴 t=(h, c, p)가 주어졌을 때, 객체 생성을 위해 헤더 h를 가지는 테이블을 찾고 해당 테이블의 헤더 값을 추출한다. 이렇게 추출된 값을 v라고 할 때, 객체 생성을 위한 후보 속성 값은 패턴 t와 v를 결합하여, (h, c, p, v)로 표현한다.

어떤 테이블이 주어졌을 때, 헤더에 대해 여러 개의 서로 다른 패턴이 적용될 수 있다. 이 경우를 패턴 충돌로 간주한다. 그림 6은 객체 생성의 예로서 테이블 헤더 Title과 Alias에 대해 각각 두 개의 패턴이 적용되어 충돌하는 것을 보이고 있다.

본 논문은 패턴 충돌 해소를 위해 같은 테이블에 등장하는 모든 헤더들의 패턴들을 연계하여 적합한 패턴을 선택한다. 이를 위해 주어진 테이블의 모든 헤더들은 동일한 개념의 객체 혹은 객체들의 속성을 표현한다고 가정한다. 패턴 충돌은 동일한 개념을 가지는 패턴 그룹들 중 가장 접합한 것을 선택함으로서 해소된다. 즉, 그림 6과 같이 dpedia:Sport, dbpedia:Book, 그리고 dbpedia:Person 각각을 가지는 패턴들의 그룹을 찾고, 각 그룹에 점수를 부여하여 가장 적합한 패턴 그룹을 선택한다. 특정 그룹을 선택함으로써 패턴들의 충돌은 해소된다.

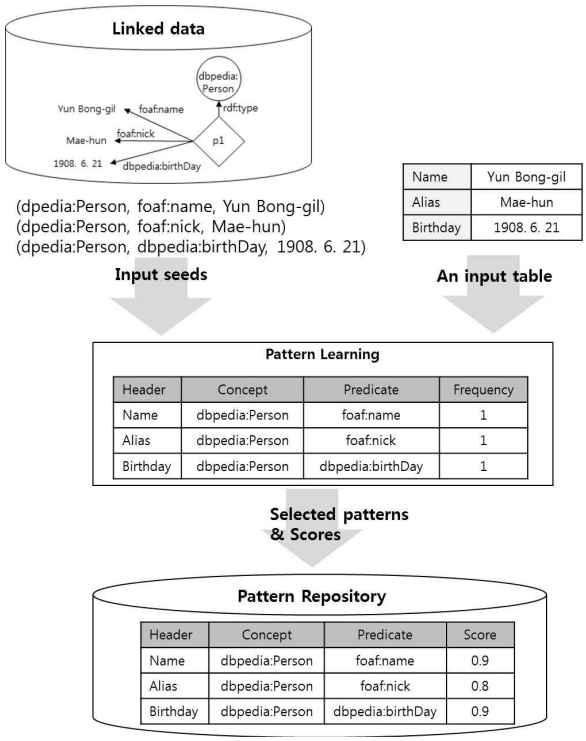

그림 5. 패턴 학습 예
Fig. 5. An example of pattern learning

주어진 테이블 헤더들의 집합을 H라 하고, 개념 c를 가지는 패턴들의 집합을 C라고 할 때, 패턴 집합 C에 대한 점수는 다음의 식 2와 같이 계산한다.

$$Score(C) = \frac{\sum_{c \in C} c.score \cdot I(c.h \in H)}{|H|} \quad (2)$$

$I(\cdot)$는 입력이 참인 경우 1을, 그렇지 않은 경우 0을 반환한다. c.score는 패턴 생성 단계에서 얻은 패턴의 점수를 의미하고, c.h는 패턴 c의 헤더를 의미한다.

그림 6에서 dbpedia:Sport를 가지는 패턴 그룹들의 매칭 경우가 많기 때문에 높은 점수를 받게 된다. 선택된 패턴 그룹을 적용하여 추출된 속성 값을 이용하여 객체를 바로 생성할 수 있다. 생성된 객체는 링크드 데이터에 저장된다.

4. 실험

실험을 통해 제안한 패턴 충돌 해소 방법이 링크드 데이터 생성 성능에 미치는 효과를 조사하였다. 실험을 위해 링크드 데이터로서 한국어 DBPedia를 사용하고 한국어 위키피디아의 infobox를 테이블 정보로 활용하였다. DBPedia는 위키피디아의 infobox 정보를 활용하여 구축되었기 때문에 이미 다량의 infobox 정보를 포함하고 있다. 실험을 위해 일부 인스턴스들을 seed 객체로 활용하고 나머지는 성능평가를 위한 gold standard로 사용하였다. 총 3개 개념, School, Company, 그리고 Film에 대해 각 4,128개, 4,220개, 5,549개의 인스턴스들을 대상으로 실험을 진행하였다. 이들 각 개념에 대해 20개의 인스턴스를 seed로, 나머지는

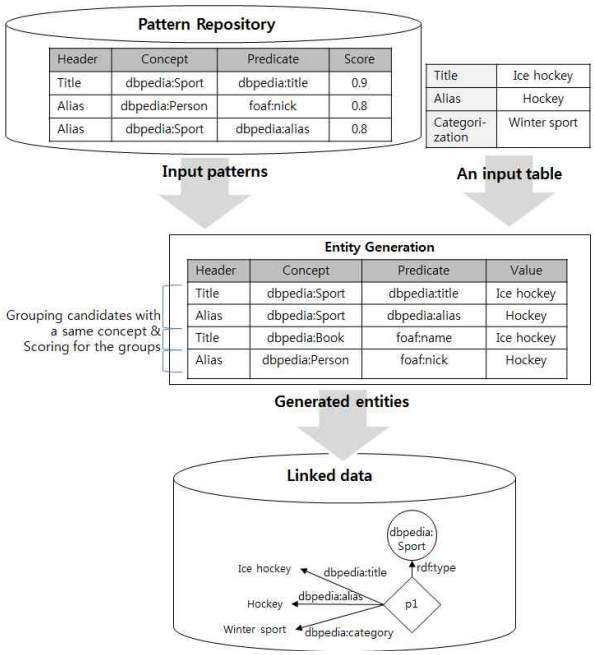

그림 6. 객체 생성 예
Fig. 6. An example of entity generation

테스트 데이터로 활용하였다. 테스트 객체들의 속성 및 대응하는 값의 개수는 총 26,339 개이다. Infobox 태그의 특징을 이용한 휴리스틱 규칙을 통해 총 229,603 개의 테이블 데이터를 추출하였다.

평가를 위해 정확률과 재현율, F1 점수를 측정하였다. 정확률은 추출한 객체의 속성 중 정확하게 추출된 객체 속성의 퍼센트이고, 재현율은 추출 대상 객체의 속성 중 정확하게 추출된 객체 속성의 퍼센트이다. 객체 속성 추출의 정확성 여부는 추출된 속성값, 대응하는 속성과 개념이 테스트 데이터에 존재하는지 여부로 판단한다.

표 1은 제안한 방법과 CSEAL-like 방법의 성능을 비교한 결과이다. 표 1에서 보는 바와 같이 모든 경우, 제안한 방법의 재현율과 F1 점수가 CSEAL-like 방법을 월등히 앞선다. 특히, 평균적인 성능을 비교하면, 제안한 방법이 CSEAL-like방법과 비슷한 정확률을 보이면서 높은 재현율을 보이고 있다. 따라서, 제안한 방법을 통해 CSEAL-like 방법과 유사한 정확률로 높은 커버리지를 갖는 링크드 데이터를 생성할 수 있다.

표 1에서 정확률이 10%이상 우세한 경우를 별표(*)로 표시하였다. 표 1에서 보는 바와 같이, 두 가지 방법이 비등하게 우세한 경우를 보이고 있다. 제안한 방법은 CSEAL-like로 찾을 수 없는 속성 정보도 고려하기 때문에 상대적으로 정확률이 낮은 경우에도 월등히 높은 재현율을 보이고 있다. 제안한 방법이 상대적으로 높은 정확률을 보이는 경우는 CSEA-like로 찾을 수 없는 속성 정보를 성공적으로 찾았음을 의미한다. 예를 들어, foaf:homepage의 경우, '웹 사이트, '홈페이지' 등 모든 대응하는 헤더의 표현들이 개념 Company, School에 대해 동일하였다. 이러한 경우, CSEAL-like 방법은 해당 패턴들을 제거하기 때문에 속성 값을 추출하지 못한다.

제안한 방법으로 패턴 충돌 해소가 어려운 경우는 크게 두 가지이다. 우선 많은 경우, 제안하는 방법은 실험에서 고

표 1. 제안한 방법과 CSEAL-like 방법 비교
Table 1. Comparison of the proposed method and the CSEAL-like method

Concept	Predicate	Proposed Method			CSEAL-like method		
		Pre.	Rec.	F1	Pre.	Rec.	F1
School	address	85.2	**100**	**92.0**	**99.6***	35.5	52.4
Film	basedOn	82.1	**82.9**	**82.5**	**82.8**	11.7	20.5
Film	cinema tography	94.2	**98.7**	**96.4**	**95.0**	6.1	11.5
Film, Company	country	**65.3***	**61.6**	**63.4**	55.0	27.6	36.7
Film	director	**49.6***	**93.1**	**64.7**	4.3	0.2	0.4
Film	distributor	**88.5***	**62.2**	**73.0**	78.0	8.2	14.8
Film	editing	**92.4**	**95.9**	**94.1**	85.7	4.1	7.9
Company	formation Year	34.7	**100**	**51.5**	**46.2***	24.0	31.6
Company	foundedBy	29.8	**100**	**45.9**	**33.0**	12.9	18.6
Company	industry	**77.9**	**99.8**	**87.5**	69.6	12.1	20.6
Film	language	**77.4***	**90.0**	**83.2**	60.1	31.2	41.1
Company	location	**95.6**	**98.4**	**96.9**	92.9	7.3	13.6
School	motto	97.5	**98.9**	**98.2**	**97.6**	98.0	97.8
Film	music Composer	45.0	**96.6**	**61.4**	**68.1***	8.4	15.0
Film	narrator	35.0	**100.0**	**51.9**	**100.0***	14.3	25.0
School	numberOf Students	84.4	**87.1**	**85.7**	**94.7***	58.1	72.0
Company	parent Company	**63.5**	**92.4**	**75.3**	58.3	7.9	13.9
Company	predecessor	**78.0***	**67.8**	**72.6**	52.0	11.3	18.6
School	president	88.3	**99.3**	**93.5**	**91.2**	34.2	49.8
Company	regionServed	**89.4**	**71.7**	**79.6**	88.5	17.8	29.7
Company	service	90.1	**100**	**94.8**	**94.1**	21.9	35.6
School	staff	92.9	**100**	**96.3**	**94.4**	65.4	77.3
Film	starring	94.6	**97.9**	**96.2**	**98.1**	13.0	22.9
Company, School	type	40.9	**97.8**	**57.7**	**86.2***	35.8	50.6
Company, School	foaf: homepage	**30.5***	**69.7**	**42.4**	0	0	0
Average		72.1	**90.5**	**77.5**	**73.0**	22.7	31.1

려되지 않은 개념들의 객체들을 실험에 사용된 개념들로의 맵핑을 시도한다. 예를 들어, 속성 type의 경우, 실험에서 고려하지 않은 University에 해당하는 객체들의 속성이기도 하다. 따라서, 이들 객체를 Company나 School의 객체로 맵핑하는 오류가 있었다. 향후 추출할 객체의 개념을 늘임으로써 정확률의 변화를 조사해 볼 계획이다. 다른 하나는 특정 객체가 둘 이상의 속성들에 대한 값이 될 수 있는 경우이다. 예를 들어, '톰 행크스'는 어떤 영화의 starring일 수도 있고, narrator일 수도 있다. 이러한 경우 통계적으로 우세한 경우를 선택하기 때문에 패턴 충돌 해소의 정확률을 떨어뜨릴 수 있다.

5. 결론 및 향후 연구

패턴 기반의 지식 추출 및 학습 방법들은 본질적으로 패턴 충돌이라는 문제를 안고 있다. 본 논문은 이러한 패턴 충돌 문제를 해소하면서 테이블로부터 링크드 데이터 생성을 위한 새로운 방법을 제안하였다. 제안한 방법은 기본적으로 하나의 테이블은 단일한 개념의 객체 혹은 객체들의 정보를 표현하고 있다고 가정한다. 이러한 가정에 따라 주어진 테이블로부터 데이터의 개념과 직접 연결되는 속성들의 정보를 통합하여 추출한다. 제안한 방법은 링크드 데이터의 개념 및 관계 정보와 테이블의 데이터를 연계함으로써 신뢰할 만한 객체를 생성한다. 실험에서 충돌 해소 방법이 적용되지 않은 경우와 비교하여 비슷한 정확률을 유지하면서 월등히 높은 재현율을 보였다.

제안한 패턴 충돌 해소 방법은 bootstrapping을 통한 링크드 데이터 생성 알고리즘에 적용할 수 있도록 고안되었다. 향후 연구로 제안한 방법을 bootstrapping 알고리즘에 적용하고 실제 웹상의 테이블들을 대상으로 실험할 계획이다.

References

[1] Cafarella, M.J., Halevy, A.Y.,Wang, Z.D.,Wu, E., and Zhang, Y., "Webtables: exploring the power of tables on the web," *PVLDB*, vol. 1, no. 1, pp. 538-549, 2008.

[2] Yoon, S.-Y., "A Study on National Linking System Implementation based on Linked Data for Public Data," KISSE, vol. 30, no. 1, pp. 259-284, 2013.

[3] Limaye, G., Sarawagi, S., and Chakrabarti, S., "Annotating and searching web tables using entities, types and relationships." *Proceedings of VLDB*, pp. 1338-1347, 2010.

[4] Carlson, A., Betteridge, J., Wang, R. C., Hruschka Jr, E. R., and Mitchell, T. M.. "Coupled semi-supervised learning for information extraction," *Proceedings of WSDM*, pp. 101-110, 2010.

[5] Mulwad, V., Finin, T., and Joshi, A.. "Semantic Message Passing for Generating Linked Data from Tables." *Proceedings of ISWC*, pp. 363-378, 2013.

[6] Wang, R. C., and Cohen, W. W., "Character-level analysis of semi-structured documents for set expansion." *Proceedings of EMNLP*, pp. 1503-1512. 2009.

[7] Kang, S.-J., "English-Korean Cross-lingual Link Discovery Using Link Probability and Named Entity Recognition," *KIIS* vol. 23, no. 3, pp. 191-195, 2013.

[8] Kang, S.-J. and Kang I.-S., "Generalization of Ontology Instances Based on WordNet and Google," *KIIS* vol. 19, no. 3, pp. 363-370, 2009.

[9] Chang, M.-S., "A Study on Focused Crawling of Web Document for Building of Ontology Instances," *KIIS* vol. 19, no. 3, pp. 363-378, 2008.

[10] Hurst, Matthew. "Towards a theory of tables." *IJDAR*, vol 8, no 2, pp. 123-131, 2006.

[11] Wang, J., Wang, H., Wang, Z., & Zhu, K. Q. "Understanding tables on the web." *Proceedings of Conceptual Modeling*, pp. 141-155, 2012.

[12] Pantel, P., and Pennacchiotti, M., "Espresso: Leveraging generic patterns for automatically harvesting semantic relations." *Proceedings of ACL*, pp. 113-120, 2006.

[13] Nakashole, N., Theobald, M., & Weikum, G., "Scalable knowledge harvesting with high precision and high recall." *Proceedings of WSDM*, pp. 227-236, 2011.

저 자 소 개

한용진(Yong-Jin Han)
2006년 : 경북대학교 컴퓨터공학과 학사
2008년 : 경북대학교 컴퓨터공학과 석사
2009년~현재 : 경북대학교 대학원 컴퓨터학부 박사과정

관심분야 : 시멘틱 웹, 자연어 처리, 기계학습
E-mail : yjhan@sejong.knu.ac.kr

김권양(Kweon Yang Kim)
1983년 : 경북대학교 전자공학과 학사
1990년 : 경북대학교 전자공학과 석사
1998년 : 경북대하교 컴퓨터공학과 박사
1983년~1988년 : ETRI 연구원
1999년~2000년 : University of Central Florida 방문 교수
1991년~현재 : 경일대학교 컴퓨터공학부 교수

관심분야 : 시멘틱 웹, 한글공학
Phone : +82-53-850-7287
Fax : +82-53-850-7609
E-mail : kykim@kiu.ac.kr

박세영(Se Young Park)
1980년 : 경북대 전자공학과 학사
1992년 : 한국가학기술원 전산학과 석사
1999년 : 프랑스 파리 7대학 전산학과 박사
1982년~2000년 : ETRI 책임연구원
2003년~2005년 : 정보통신연구진흥원 전문위원
2005년~현재 : 경북대학교 컴퓨터공학과 교수

관심분야 : 시멘틱 웹, 자연어 처리, 정보 검색
Phone : +82-53-950-6551
Fax : +82-53-950-2122
E-mail : seyoung@knu.ac.kr