지식베이스 확장을 위한 행렬 분해 모델

김지호○, 남상하, 최기선

한국과학기술원
{hogajiho, nam.sangha, kschoi}@kaist.ac.kr

Matrix Factorization Models for Knowledge Base Population

Jiho Kim○, Sangha Nam, Key-Sun Choi
KAIST

요 약

지식베이스의 목표는 세상의 모든 지식을 데이터베이스화 하는 것이지만 지식 획득 능력의 부족으로 항상 지식 부족 문제에 시달린다. 지식 획득은 주로 웹 상에 있는 자연언어문장을 지식화 하는 외부적인 지식 획득을 통해 이루어지지만, 지식베이스 내부에서 지식을 확장해 나가는 방법에 대해서는 연구가 소홀히 이루어지고 있다. 따라서 본 논문에서는 내부적인 지식 획득을 위한 지식베이스 행렬 분해 모델을 소개한다. 본 논문에서 소개하는 방법은 지식베이스를 행렬로 변환한 뒤 행렬 분해 모델을 통해 새로운 지식에 대한 신뢰도를 점수화하는 방법이다. 본 논문에서 소개한 방법의 우수성과 실효성을 입증하기 위해 한국어 지식베이스인 한국어 디비피디아(2016-10)를 대상으로 본 모델의 정확도 측정 실험 결과를 소개한다.

주제어: 지식베이스, 지식베이스 확장, 행렬 분해, 지식 획득

1. 서론

웹과 인터넷의 등장과 함께 세상에는 방대한 양의 정보가 공유되기 시작했다. 방대한 정보를 효율적으로 관리하고 사용하려면 기계가 해석할 수 있도록 정보를 정제하는 과정이 필요하다. 이를 위해 정보를 구조화하고 데이터베이스화 시킨 지식베이스(knowledge base)에 대한 연구가 활발히 진행되고 있다. 지식베이스는 정보를 트리플(triple) 구조로 저장한다. 트리플이란 두 개체(entity)간의 관계(relation) 정보를 <주어 개체 - 관계 - 목적어 개체>와 같이 표현하는 구조이다. 예를 들면 "대한민국의 수도는 서울이다." 라는 정보를 트리플 형태로 표현하면 <대한민국 - 수도 - 서울>과 같이 표현할 수 있다.

지식베이스가 유용하게 쓰이기 위한 조건은 크게 두 가지가 있다. 첫 번째는 포함하고 있는 정보의 범위가 넓어야 하는 것이고 두 번째는 포함하고 있는 정보의 정확성이 높아야 하는 것이다. 즉 지식베이스의 유용성을 높이기 위해서는 지속적으로 지식베이스에 새롭고 정확한 지식을 추가하는 지식베이스 확장(knowledge base population)이 필요하다. 지식베이스 확장을 위해서는 새롭게 추가할 트리플을 습득해야 하고, 이에 선행하여 새로운 지식을 알아내는 지식 획득(knowledge acquisition)이 이루어져야 한다.

지식 획득은 새로운 지식의 출처에 따라 지식베이스 내부에서 새로운 지식을 획득하는 내부적인 지식 획득(interior knowledge acquisition)과 지식베이스 외부에서 새로운 지식을 획득하는 외부적인 지식 획득(exterior knowledge acquisition)로 나눌 수 있다. 지식 획득에 관한 연구는 주로 웹에 존재하는 자연언어문장을 읽고 지식을 추출하는 관계추출(relation extraction)에 집중되어 있다. 관계추출에 관한 연구에는 TAC Knowledge Base Population[1][2][3], OpenIE[4], OLLIE[5] 등이 있다. 이들은 영어 도메인에서 자연언어문장을 읽어들여 개체명 간의 관계를 나타내는 트리플을 추출하는 문제에 집중하였다.

지식베이스 확장의 효율성을 높이기 위해서는 내부적인 지식 획득이 동반되어야 한다. 내부적인 지식 획득에 관한 연구로는 귀납적 논리 프로그래밍(inductive logic programming)에 기반한 AMIE[10] 등이 있다. 이들은 지식베이스로부터 논리적 법칙(logical rules)들을 추출하여 새로운 지식을 알아내는 문제에 집중하였다. 하지만 아직 내부적인 지식 획득에 있어 행렬 분해 모델을 이용한 연구는 없다. 따라서 본 논문에서는 내부적인 지식 획득 방법인 KBMF(Knowledge Base Matrix Factorization)을 소개한다. 본 방법은 지식베이스에 저장되어 있는 정보만을 이용하여 새로운 지식에 대한 신뢰도를 점수화하는 방법이다. KBMF를 통해 지식베이스에 새로운 트리플을 추가할 수 있으며, 추가하려는 트리플의 신뢰성까지 측정이 가능하다. 또한 본 방법은 지식베이스의 행렬화와 행렬 분해(matrix factorization) 모델을 분리해서 설계하였기 때문에 추후 관계추출 결과와 지식베이스를 통합하여 분석하는 방향으로 시스템을 확장해 나갈 수 있다.

본 논문은 내부적인 지식 획득에 있어서 본 논문에서 소개한 KBMF가 효과적임을 보이는 것에 중점을 둔다. 한국어 디비피디아(Korean DBpedia)[6]에 본 논문의 행렬 분해 방법을 적용하여 실험해 봄으로써 실효성을 증명하고자 한다.

2. 지식베이스 행렬 분해 모델(KBMF)

KBMF는 행렬화 단계와 행렬 분해 단계, 그리고 상위 트리플 추출 단계로 구성된다. 모델의 흐름도는 그림 1과 같다. KBMF는 지식베이스에 저장된 트리플들을 행렬화하고, 행렬 분해 모델을 통해 개체쌍(entity tuple)과

관계 각각에 대한 특징 벡터를 학습한 뒤 새로운 트리플들이 참일 확률을 계산하여 지식베이스 행렬을 최적화한다. 최적화된 지식베이스 행렬 로부터 학습 데이터를 제외한 나머지 중 높은 확률을 가진 트리플들을 추출하여 새로운 지식을 추출하게 된다.

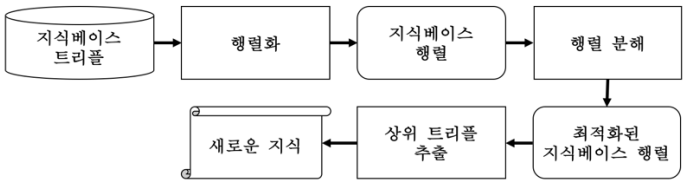

그림 1 KBMF 흐름도

2.1. 지식베이스 행렬화

지식베이스에는 <주어 개체 - 관계 - 목적어 개체>로 구성된 트리플 형태로 지식이 저장되어 있다. 지식베이스 행렬화란 <주어 개체, 목적어 개체> 로 이루어진 개체쌍을 행으로, 관계를 열로 가진 행렬로 트리플들을 표현하는 작업이다. 그림 2는 두 개의 트리플을 행렬화 하는 예시를 나타내고 있다. 관계 "장군(X,Y)"는 "X의 장군 Y"라는 관계이고, "수도(X,Y)"는 "X의 수도는 Y"라는 관계를 나타낸다.

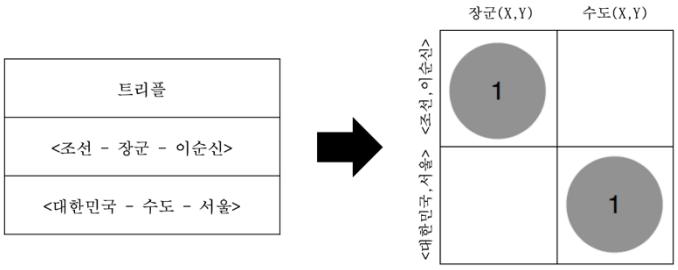

그림 2 트리플 행렬화의 예시

행렬화를 통해 지식베이스에 등록된 트리플은 1이고, 나머지 칸(slot)은 모두 0인 행렬 K를 얻을 수 있다.

2.2. 행렬 분해

행렬 분해 모델의 목표는 2.1에서 얻은 지식베이스 행렬 K의 근사 행렬 K'을 구하는 것이다. 2.2.1에서는 근사에 사용할 목적 함수(objective function)에 대해 소개한다. 2.2.2에서는 행렬 분해에 사용할 자연 매개 변수(natural parameter)들에 대해 설명한다. 2.2.3에서는 행렬 분해 알고리즘에 대해 설명한다.

2.2.1. 목적 함수

본 논문의 행렬 분해 모델에서 사용하는 목적 함수는 BPR[7]에서 사용한 목적 함수와 동일하다. 상품 추천 시스템에서 행과 열을 이루는 고객과 상품은 본 논문에서 각각 개체쌍과 관계에 대응된다. 어떤 개체쌍 $t = < sbj, obj >$가 관계 r_i을 관계 r_j보다 선호할 확률(r_i이 성립할 확률이 r_j가 성립할 확률보다 높을 확률을 의미한다)은 식(1)과 같이 정의한다.

$$p(r_i >_t r_j|\Theta) := \sigma(\hat{x}_{tij}(\Theta)) \quad \text{식(1)}$$

σ 는 로지스틱 시그모이드 함수(logistic sigmoid function)이며, Θ는 모델의 자연 매개 변수이다. $\hat{x}_{tij}(\Theta)$는 결과값이 실수인 함수이며, 행렬 분해 모델에 따라 달라지는 함수가 된다. 우리의 목적은 학습 데이터에 속한 트리플 (t, r_i)와 속해 있지 않은 트리플 (t, r_j)에 대해 $p(r_i >_t r_j|\Theta)$를 최대화하는 것이다. $\hat{x}_{tij}(\Theta)$는 식(2)와 같이 정의된다. $\hat{x}_{tij}(\Theta)$에서 변수 Θ는 앞으로 편의상 생략한다.

$$\hat{x}_{tij} := \hat{x}_{ti} - \hat{x}_{tj} \quad \text{식(2)}$$

여기서 $\hat{x}_{ti}$는 행렬 분해 모델에서 최종적으로 구하고자 하는 K'의 t번째 행, i번째 열의 값이다. 이는 2.2.2에서 중점적으로 다루게 될 것이다. 우리가 행렬 분해 모델에서 최대화하고자 하는 목적 함수는 식(3)과 같다. 이 목적 함수가 의미하는 바는 '학습 데이터에 부여하는 확률 값과 나머지 데이터에 부여하는 확률 값의 차이'다.

$$Obj = \sum_{(t,i,j) \in D} ln\, \sigma(\hat{x}_{tij}) - \lambda_\Theta \|\Theta\|^2 \quad \text{식(3)}$$

λ_Θ 는 모델에 따른 정규화 변수(regularization parameter)다. D의 정의는 식(4)와 같다.

$$D := \{(t,i,j) | (t,r_i) \in KB, (t,r_j) \notin KB\} \quad \text{식(4)}$$

2.2.2. 자연 매개 변수

본 논문에서는 $\hat{x}_{ti}$에 대한 총 5가지의 모델을 사용한다. $\hat{x}_{ti}$가 의미하는 것은 최적화된 지식베이스 행렬의 slot값이며, 개체쌍 t와 관계 i로 이루어진 트리플에 모델이 부여하는 확률 값이다. 각 $\hat{x}_{ti}$ 모델은 행렬 K를 분해하는 서로 다른 방법들이기도 하다. 모델들은 Riedel et al.(2013)[8]을 참고했음을 밝힌다. 각 모델의 이름, $\hat{x}_{ti}$의 정의, 자연매개변수 목록을 표 1에 정리하였다. 편의상 관계 r_i는 i로 표현하였다. f-model(latent feature model)은 개체쌍의 특성 벡터 $\mathbf{w}$와 관계의 특성 벡터 $\mathbf{h}$의 선형 결합으로 $\hat{x}_{ti}$ 를 정의한다. n-model(neighborhood model)은 관계 간의 weight matrix R을 정의한 뒤 어떤 트리플 (t,i)에 대해 $(t,j) \in KB$인 모든 관계 j를 찾고 관계 i와 j사이의 가중치 값 $r_{i,j}$의 합으로 $\hat{x}_{ti}$를 정의한다. nf-model은 n-model과 f-model의 합으로 정의한다. e-model(entity model)은 개체쌍 $t = (e_1, e_2)$ 에 대해 각 개체별 특성 벡터 e_{e_1}, e_{e_2}를 정의하여 이를 관계의 특성 벡터 d_1, d_2와의 선형 결합으로 $\hat{x}_{ti}$를 정의한다. d_1은 관계의 주어 개체 자리의 특성 벡터를, d_2는 목적어 개체 자리의 특성 벡터를 의미한다. nfe-model은 (n,f,e)-model들의 합으로 정의한다.

모델	$\hat{x}_{ti}$	Θ
f-model	$\hat{x}_{ti} = \sum_{k}^{dim} w_{t,k} h_{i,k}$	W, H
n-model	$\hat{x}_{ti} = \sum_{(t,j)\in KB} r_{i,j}$	R
nf-model	$\hat{x}_{ti} = \sum_{k}^{dim} w_{t,i} h_{t,i} + \sum_{(t,j)\in KB} r_{i,j}$	W, H, R
e-model	$\hat{x}_{ti} = \sum_{a=1}^{2} \sum_{k}^{dim} d_{a,k} e_{e_a,k}$	D, E
nfe-model	$\hat{x}_{ti} = \sum_{k}^{f\_dim} w_{t,i} h_{t,i} + \sum_{(t,j)\in KB} r_{i,j} + \sum_{a=1}^{2} \sum_{k}^{e\_dim} d_{a,k} e_{e_a,k}$	W, H, R, D, E

표 1 행렬 분해 모델

2.2.3. 행렬 분해 알고리즘

본 절에서 설명할 행렬 분해 알고리즘은 1과 0으로 이루어진 행렬 K를 입력 받아 목적 함수를 최대화하도록 확률적 경사 하강법(Stochastic Gradient Descent)을 통해 모델 별로 2.2.2에서 정의한 자연매개변수들을 최적화하여 반환한다. 최적화한 자연매개변수들을 다시 조합하여 K의 근사 행렬 K'를 출력할 수 있다. 그림 3은 행렬 분해 알고리즘의 의사코드(pseudocode)이다.

알고리즘 1 행렬 분해 알고리즘

```
1:  procedure OptimizeKBMF(Triples,Θ,model)
2:    initialize Θ
3:    repeat for batch_size
4:      randomly draw (t,i,j)from Triples
5:      Θ ← Θ + α(∂Objective Function/∂Θ)
6:    return Θ
```

그림 3 행렬 분해 알고리즘

본 알고리즘은 지식베이스에서 뽑아낸 트리플들의 리스트(Triples)와 자연매개변수(Θ), 그리고 모델 이름(model)을 입력으로 받아 학습된 자연매개변수를 반환한다.알고리즘의 핵심은 트리플 리스트로부터 트리플 하나 (t,i) 를 무작위로 추출하고, 리스트에 속하지 않은 트리플 (t,j)를 무작위로 추출하여(Line 4) 이를 (t,i,j)로 합쳐 확률적 경사 하강법을 진행하는 것이다(Line 5). α는 학습 정도(learning rate)이며, $\frac{\partial Objective\ Function}{\partial \Theta}$은 모델에 따라 달라진다. 정해진 반복 횟수(batch_size)만큼 샘플 추출 과 확률적 경사 하강법을 반복하여 최적화된 Θ를 반환하게 된다.

3. 실험

본 논문에서 소개한 KBMF의 평가를 위해 한국어 디비피디아를 대상으로 두 가지 실험을 진행하였다. 2.2에서 소개한 다섯 가지 모델들에 대한 성능평가 실험을 3.1에서, 추출한 지식에 대한 정밀도(precision) 평가 실험을 3.2에서 소개한다. 두 실험에서 사용한 데이터셋은 표 2에 정리되어 있다.

항목	개수
대상 관계 수	59
대상 관계를 3개 이상 가진 개체쌍	1217
트리플 수	3702

표 2 실험 데이터셋

3.1 모델 별 성능 평가

본 실험의 목적은 2.2에서 소개한 다섯 가지 행렬 분해 모델의 성능을 평가하는 것이다. 평가 방법으로는 one out evaluation scheme을 사용하였다. 각 개체쌍 별로 가지고 있는 관계 하나씩을 무작위로 골라 실험 데이터로 분리하고, 원래 데이터셋에서 실험 데이터를 제외한 나머지를 학습 데이터로 사용하였다. 학습 데이터를 입력으로 넣어 지식베이스 행렬에 행렬 분해 모델을 적용한 뒤 식(5)와 같이 AUC(area under the ROC curve)를 계산하였다.

$$AUC = \frac{1}{|T|}\sum_{t}\frac{1}{|E(t)|}\sum_{(i,j)\in E(t)}\delta(\hat{x}_{ti} > \hat{x}_{tj}) \quad \text{식(5)}$$

$E(t)$는 식(6)과 같이 정의한다.

$$E(t) := \{(i,j) | (t,i) \in S_{test} \wedge (t,j) \notin (S_{train} \cup S_{test})\} \quad \text{식(6)}$$

AUC가 높을수록 모델의 신뢰도가 높다고 할 수 있다. 각 트리플에 대해 무작위로 점수를 매기는 모델의 AUC값은 0.5에 근접할 것이며, AUC의 최대값은 1이다.

실험에서 사용한 hyperparameter들은 nfe-model을 기준으로 가장 좋은 성능을 보이는 값들이며, 표 3에 정리되어 있다. 기준을 nfe-model로 삼은 이유는 hyperparameter에 관계 없이 다른 모델들보다 항상 높은 성능을 보여주었기 때문이다. f_dim 은 W, H 의 dimension을 의미하고, e_dim은 D, E 의 dimension을 의미한다.

Hyperparameter	Value
f_dim	30
e_dim	3
batch_size	len(train_set) * 30
learning rate	0.03

표 3 Hyperparameters

각 모델의 AUC를 측정한 결과는 표 4에 정리되어 있다. 모델 별로 각각 10번씩 실험하여 얻은 AUC값의 평균을

냄으로써 측정하였다.

모델	AUC
f-model	0.6022
n-model	0.8825
nf-model	0.8770
e-model	0.5551
nfe-model	0.9530

표 4 모델별 AUC 측정 결과

다섯 모델 중에서 nfe-model이 0.9530의 가장 높은 성능을 보여주었다.

3.2 추출한 지식의 정밀도 평가

본 실험의 목적은 KBMF가 실제로 새로운 지식을 잘 추출해낼 수 있는지 알아보기 위함이다. 가장 높은 성능을 보였던 nfe-model을 적용한 행렬 분해를 통해 얻은 최적화된 지식베이스 행렬에서 학습 데이터를 제외한 나머지 중 점수가 가장 높은 100개의 트리플을 뽑아 실제로 맞는 트리플인지 사람이 직접 평가하였다. 표 5에 평가 결과를 관계 별로 정리하였다.

관계	트리플 수	참인 트리플	거짓인 트리플	정밀도
nationality	22	20	2	91%
deathPlace	19	4	15	21%
birthPlace	14	5	9	36%
city	14	14	0	100%
writer	9	1	8	11%
club	7	7	0	100%
director	6	1	5	17%
team	3	2	1	67%
producer	2	0	2	0%
routeEnd	2	1	1	50%
routeStart	1	0	1	0%
managerClub	1	1	0	100%
total	100	57	43	57%

표 5 상위 100개 트리플 정밀도 측정 결과

추출한 트리플 100개 중 57개가 참인 트리플로 평가되었다. 관계의 종류에 따라 정밀도가 큰 차이로 변화하는 것을 확인할 수 있다. 추출된 트리플 수가 5개 이상인 관계 중 nationality, city, club은 90%가 넘는 정밀도를 보인 반면, deathPlace, birthPlace, writer, director의 경우 40% 미만의 낮은 정밀도를 보였다.

높은 정밀도를 보였던 city가 추출된 개체쌍의 경우 해당 개체쌍의 학습 데이터에 routeStart, routeEnd, LocatedIn이 포함되어 있었다. 예를 들어 city가 추출된 개체쌍 <대구_도시철도_1호선 - 대구광역시>의 경우 나머지 세 관계가 학습 데이터에 포함되어 있었다. club의 경우 managerClub, team, position이 학습 데이터에 포함되어 있었고, nationality의 경우 birthPlace, deathPlace, position이 학습 데이터에 포함되어 있었다. 학습 데이터의 세 관계로부터 추출된 관계를 추론할 수 있는 경우에 높은 정밀도를 보여주었다고 판단할 수 있다. nationality에서 추출된 거짓인 트리플 <윌리엄_카를로스_윌리엄스 - nationality - 뉴저지_주>의 경우 '뉴저지_주'가 국가가 아니어서 nationality 관계가 성립할 수 없었기 때문이다.

낮은 정밀도를 보였던 deathPlace가 추출된 개체쌍의 경우 해당 개체쌍의 학습 데이터에 birthPlace, country, nationality가 포함되어 있었다. 이는 대부분의 인물 개체에서 birthPlace, country, deathPlace, nationality가 모두 동일하기 때문이다. 개체쌍이 birthPlace, country, nationality 관계를 가지는데 deathPlace를 가지지 않는 경우는 대부분 해당 인물 개체가 다른 곳에서 사망하였기 때문인데, 이 패턴은 학습되지 않았기 때문에 deathPlace 관계가 추출되었다고 판단된다. deathPlace 관계를 맞게 추출한 트리플 <원균 - deathPlace - 조선>의 경우 지식베이스에 deathPlace가 누락되어 있었다. birthPlace의 경우도 deathPlace와 유사하게 country, nationality, deathPlace가 학습 데이터에 있는 경우에 추출되었다. 낮은 정밀도를 보이는 이유 또한 동일하다. birthPlace, country, nationality, deathPlace 관계를 모두 가지는 개체쌍이 많지만, 출생지와 사망지가 다른 인물들에 대한 트리플을 주로 추출하였기 때문에 낮은 정밀도를 보인다고 해석된다.

4. 관련 연구

행렬 분해란 행렬 K를 행렬 T와 R의 곱으로 근사하는 것을 말한다. 행렬 분해는 이미지 처리, 문서 분류, 상품 추천 시스템 등에서 쓰이고 있다. 특히 상품 추천 시스템에서 행렬 분해를 이용한 연구가 많이 수행되었다.

상품 추천 시스템이란 특정 상품 목록에 대해 개인화된 순위를 매기는 문제를 푸는 작업을 의미한다. BPR(Bayesian Personalized Ranking from Implicit Feedback)[7]은 베이지안 분석 기법(Bayesian Analysis)에 의거한 순위 매김에 최적화 되어 있는 목적 함수를 처음 제안하고 확률적 경사 하강법을 이용한 행렬 분해 알고리즘을 선보였다. 제안된 알고리즘은 10,000명의 고객과 4,000개의 상품에 대한 426,612회의 구매가 기록된 로스만 데이터(Rossmann Dataset)에 대해 AUC 값이 최대 0.92까지 올라감을 보였다.

최근에는 관계추출 문제에서 행렬 분해를 적용하려는 연구들이 활발히 이루어지고 있다. 관계추출 문제에서 행렬 분해를 통해 얻고자 하는 것은 개체쌍과 관계 각각의 저차원 벡터 임베딩(low-dimensional vector embedding)이다. 특정 트리플에 부여되는 점수는 해당 트리플의 개체쌍 특성 벡터와 관계 특성 벡터의 선형결합으로 나타내어지게 된다. BPR[7]에서 선보인 목적 함수를 자연언어처리 분야에 적합하게 수정하고, 추가적인 모델들을 설계하여 행렬 분해를 영어 관계추출에 적용한 연구가 있다[8]. 이 연구에서 처음으로 제안한 유니버셜 스키마(universal schema)는 관계 추출 시에 특정한 지식베이스 스키마에 국한되지 않고 여러 지식베이스 스키마와 더불어 OpenIE[4]를 통해 자연언어문장에서 추출한

스키마까지 모두 통합된 스키마이다. 관계 추출 시에 유니버셜 스키마를 구축하고 행렬화 한 뒤 자신들이 제안한 행렬 분해 모델들(n-model, e-model, nf-model, nfe-model)로 전체 스키마를 근사하는 방법을 제안했다. Freebase와 NYTimes corpus를 데이터셋으로 한 실험에서 기존의 관계추출 방법들에 비한 성능 향상을 증명하였다. 이에 추가로 1차 논리(first-order logic)를 통합하여 고려한 행렬 분해 모델로 성능을 향상시킨 연구도 이루어졌다[9].

5. 결론

본 논문에서는 지식베이스 확장을 위한 내부적인 지식 획득 방법으로써 행렬 분해 모델 KBMF를 소개하였다. KBMF는 지식베이스에 등록된 개체쌍과 관계 각각에 대한 특징 벡터를 학습하여 새로운 지식의 순위를 매긴다. 한국어 디비피디아에 이를 적용하는 실험을 수행한 결과 본 새로운 지식의 순위를 매기는 데에 있어 높은 성능을 보인다는 사실을 확인하였다. 새로운 지식 추출에 있어서는 해당 지식의 관계 종류에 따라 정밀도에 큰 차이가 있음을 확인하였다. 낮은 정밀도를 보인 관계들의 경우 향후 자연언어문장으로부터 어휘적 관계를 추출하여 지식베이스와 통합하는 한국어 유니버셜 스키마에 본 논문에서 소개한 모델을 적용하는 연구를 통해 성능을 향상시키고자 한다. 또한 본 논문에서 소개한 방법을 적용한 지식베이스 확장 시스템 설계에 대한 연구, 그리고 다양한 언어 지식베이스에 대한 실험을 통해 본 모델이 언어의 종류에 구애 받지 않음을 증명하고자 한다.

사사

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (2013-0-00109, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발)

참고문헌

[1] https://tac.nist.gov/2017/KBP/

[2] Ji, Heng, and Ralph Grishman. "Knowledge base population: Successful approaches and challenges." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1.* Association for Computational Linguistics, 2011.

[3] Surdeanu, Mihai, and Heng Ji. "Overview of the english slot filling track at the tac2014 knowledge base population evaluation." Proc. Text Analysis Conference (TAC2014). 2014.

[4] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. Commun. ACM, 51(12):68-74

[5] Mausam, Mausam. "Open information extraction systems and downstream applications." Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. AAAI Press, 2016.

[6] http://ko.dbpedia.org

[7] Rendle, Steffen, et al. "BPR: Bayesian personalized ranking from implicit feedback." *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press*, 2009.

[8] Riedel, Sebastian, et al. "Relation Extraction with Matrix Factorization and Universal Schemas." *HLT-NAACL*. 2013.

[9] Rocktäschel, Tim, Sameer Singh, and Sebastian Riedel. "Injecting Logical Background Knowledge into Embeddings for Relation Extraction." *HLT-NAACL*. 2015.

[10] MLA Galárraga, Luis Antonio, et al. "AMIE: association rule mining under incomplete evidence in ontological knowledge bases." *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013.