새로운 개체 발견과 반복적 개체 연결에 대한 방법 연구

이민호°, 남상하, 김동환, 최기선
한국과학기술원
{pathmaker, nam.sangha, iedcon, kschoi}@kaist.ac.kr

Study of New Entity Discovery and Iterative Entity Linking

Minho Lee°, Sangha Nam, Donghwan Kim, Key-sun Choi
KAIST

요 약

개체 연결은 자연어 문장 안에서 나타난 개체를 지식베이스의 URI에 연결하는 작업이다. 그러나 지금까지는 새로운 개체를 지식베이스에 등록하여 지식베이스를 확장하려는 시도가 아직 없었다. 본 논문에서는 지식베이스에 새로운 개체를 등록하는 방법인 "개체 발견" 과정과, 이를 평가하는 방법인 "반복적 개체 연결"에 대한 순서와 실험 방법을 정의하였다. 실험 결과를 통해 개체명을 많이 등록할수록 새로운 URI를 잘 찾아내는 장점이 있지만, 기존의 개체 연결 성능에 악영향을 미쳐 적절한 개체 검증 과정이 필요함을 보였다.

주제어: 개체 연결, 개체 발견, 지식베이스 증강

1. 서론

지식베이스란 현실에 존재하는 개체와 개체 간의 관계를 모아 놓은 저장소이다. 지식베이스 내의 개체는 고유한 URI(통합 자원 식별자, Uniform Resource Identifier)를 가지고 있어, 서로 다른 개체는 다른 URI를 가진다. 개체 연결은 자연어 텍스트 내에서 개체를 나타내는 문자열을 대상으로 해당 개체가 지식베이스 내의 어떤 URI에 해당하는지 연결하는 과제로, 관계 추출, 질의 응답 등 지식베이스 관련 과제의 기반이 된다. 개체 연결은 지식베이스 내의 URI만을 대상으로 하기 때문에, 지식베이스에 저장된 URI의 개수가 많을수록 더 많은 종류의 개체를 지식베이스에 연결할 수 있다. 개체 연결에서 기존의 많은 연구는 개체 연결을 더 정확하게 하거나, 개체가 지식베이스에 존재하는지의 여부를 판단하거나, 개체 연결에 실패한 개체들을 묶어서 분류하는 작업 등을 수행하였다. 기존 연구에서는 지식베이스 내의 연결에 실패한 개체를 NIL 개체라고 칭하고, 이 중 실제로 지식베이스에 없는 개체를 Dark entity라고 칭하였다. 본 연구에서는 Dark entity를 발견하고 지식베이스의 URI를 부여하는 과정을 개체 발견이라고 명명한다. 개체 발견을 위해 개체 연결에 실패한 개체들을 군집화하고, 군집들을 등록이 가능한지 검증하고, 검증된 개체를 지식베이스에 등록하여 새로운 URI를 부여한다. 또한, 새로 등록된 URI를 실제로 개체 연결 과정에서 연결할 수 있는지, 또한 새로 등록된 개체들이 기존의 지식베이스로 학습한 모델에 얼마나 영향을 미치는지에 대한 실험을 진행하였다. 본 연구에서는 이 실험 과정을 '반복적 개체 연결'이라고 부르고, 이에 대한 방법과 실험 및 데이터 설계, 그리고 평가 방법을 제안한다.

본 논문에서 기여하는 바는 다음과 같다.

- 개체 발견을 통해 지식베이스를 확장하는 방법을 제시하였다.
- 반복적 개체 연결에 대한 실험 방법을 제시하고, 데이터를 만드는 법을 제시하였다.

2. 관련 연구

기존의 연구는 개체 연결, NIL 개체 구분, 그리고 Dark entity 관련 연구가 있다.

2.1 개체 연결

대다수의 기존 연구는 개체 연결의 성능을 올리는 방향으로 진행되어 왔다. 초기의 개체 연결은 위키피디아 등의 링크 정보를 가진 백과사전을 쓰는 방식 [1]이 있었다. 최근에는 신경망 기반의 모델을 사용하여, 개체의 임베딩을 구해 성능을 높인 것 [2]과, 개체 간의 상호참조 관계를 참고하거나 [3], 잠재적 관계를 가정 [4]하여 개체 연결의 성능을 크게 올렸다.

2.2 NIL 개체 구분

개체 연결에서 지식베이스 내의 URI로 연결하지 못한 개체를 NIL 개체라고 한다. 개체 연결의 성능을 향상하는 것이 목적인 연구와는 달리, NIL 개체를 구분 태스크는 지식베이스에 연결하지 못한 개체들 중 실제로 지식베이스 내에 없는 개체를 구분하는 것이 목적이다. TAC-KBP [5]에서는 지식베이스에 연결하지 못한 개체들을 군집화하는 것까지를 목표 문제로 삼았으나, 해당 개체가 새로운 개체인지, 아니면 단순히 연결에 실패한 것인지는 구분하지 않았다. [6]에서는 모호한 이름을 가진 개체가 지식베이스에 존재하는지의 여부를 구분하는 연구를 하였다. 마지막으로, [7]에서는 단순히 개체 연결의 임계

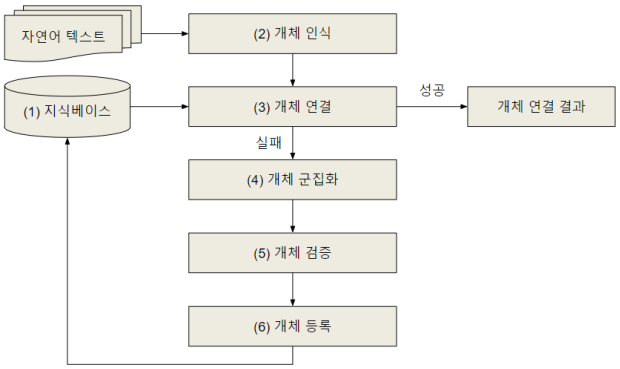

그림 1. 반복적 개체 연결의 순서도

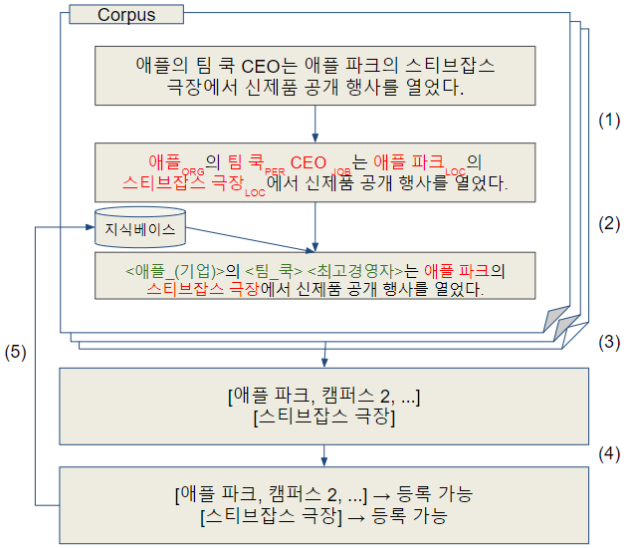

그림 2. 반복적 개체 연결의 예시. (1)-개체명 인식, (2)-개체 연결, (3)-개체 군집화, (4)-개체 검증, (5)-개체 등록을 나타낸다. 개체 등록 이후 새 텍스트가 들어온다면 “애플 파크”, “스티브잡스 극장” 등의 개체가 연결 가능하게 된다.

점수로만 구분하는 것이 아닌 여러 특질 공간에서의 분포를 바탕으로 NIL 개체를 발견하는 방식을 제시하였다.

2.3 Dark Entity

Dark entity에 대한 정의는 [8]에 처음 등장하였다. 해당 논문에서는 지식베이스에 없는 개체를 Dark entity라고 정의하였다. 또한, NewsReader 프로젝트 [9]에서는 Dark entity에 대한 해결법으로 다른 지식베이스의 개체를 참고하는 방식을 제안하였다. 본 논문에서는 다른 지식베이스에 대한 참고 없이, 새로운 문서만으로 Dark entity를 찾아내어 등록하는 것을 목표로 한다.

3. 개체 발견 및 반복적 개체 연결

본 논문에서는 ”개체 발견”이라고 부르는 새로운 개체를 등록하는 과정과, “반복적 개체 연결” 이라는 평가 방법을 제안한다. 반복적 개체 연결이란, 말뭉치에 대해 개체 연결 및 등록을 진행하여 새로운 개체를 지식베이스에 등록한 후, 비슷한 성질을 가진 다른 말뭉치를 사용하여 새로 등록된 개체를 얼마나 잘 찾아내는지 평가하는 과정이다. 이 때, 두 말뭉치는 새로 등록된 개체를 공유한다고 가정한다. 개체 발견과 반복적 개체 연결을 자동화하기 위해서는 여섯 가지의 모듈이 필요하다. 1에 본 논문에서 제시하는 반복적 개체 연결의 흐름이 나타나 있다. 이후 문단에서 각각의 모듈에 대한 정의와 목적, 예시를 다룬다.

3.1 개체명 인식

개체명 인식은 자연어 텍스트를 입력으로 받아 텍스트 내의 개체를 찾아내고, 개체의 타입을 부여하는 작업이다. 본 논문에서는 개체명 인식에서 찾아낸 개체만을 개체 연결의 대상으로 본다.

3.2 개체 연결

개체 연결은 개체명 인식에서 찾은 개체를 지식베이스 내의 URI로 연결하는 태스크이다. 개체 연결은 두 가지 작은 문제로 나눌 수 있다. 개체에 연결할 수 있는 지식베이스의 URI 후보를 뽑는 후보군 생성 과정과, 생성된 후보군에 순위를 매겨 최종적으로 연결할 URI를 찾는 후보 랭킹 과정이 있다.

3.3 개체 군집화

개체 군집화는 개체 연결 과정에서 지식베이스에 연결하지 못한 개체를 모은 뒤, 같은 대상을 나타내는 것끼리 묶는 과정이다. 개체명 인식과 개체 연결의 과정은 문서 단위에서 이루어지지만, 개체 군집화는 말뭉치 단위로 수행하여 더 많은 개체를 군집화하는 것이 목표이다. 말뭉치에 포함된 문서의 수가 많을수록 더 많은 개체가 군집화되고, 이는 개체 군집의 평균 분포를 구하는 데 도움이 된다.

3.4 개체 검증

개체 검증은 개체 군집화에서 나온 군집들이 실제로 URI를 부여받을 자격이 있는지를 검증하는 과정이다. 3.1에서 3.3의 모듈들이 완벽하다면, 위 과정을 통해 나온 군집들은 지식베이스 내에 존재하지 않으며 각각의 군집은 단일한 개체를 나타내어야 한다. 하지만 현재 영어권에서 개체 연결의 최고 성능은 93% 내외이고 [4], 개체 군집화의 최고 성능은 ARI metric 기반으로 56% 내외이다 [10]. 이 수치를 봤을 때, 해당 모듈들이 완벽하게 작동한다고 확신할 수 없기에, 앞선 모듈들을 통합적으로 검증하는 과정이 필요하다. 본 논문에서는 간단한 신경망

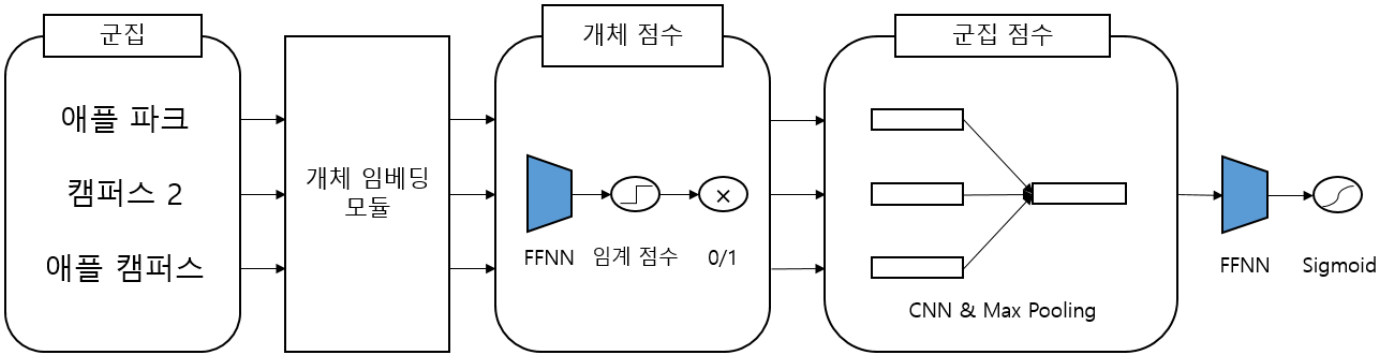

그림 3. 개체 검증 모델의 구성

기반의 검증 모델을 제작하여 실험하였다. 3.4.1에서 모델의 자세한 설명을 기술하였다.

개체 검증의 최종 목적은 개체가 아닌 것, 지식베이스에 있는 개체, 여러 대상을 나타내는 개체를 걸러내는 것이다. 그러나 개체 연결에서는 개체명과 지식베이스 내의 URI를 잇는 개체명 사전에 큰 영향을 받아, 개체명 사전에 없는 개체는 연결하기 매우 어렵다. 본 논문에서는 개체 연결의 랭킹 과정의 오류를 바로잡는 것을 목적으로 하므로, 개체명 사전에 없는 경우를 해결하는 것은 본 논문의 문제 범위로 두지 않는다.

3.4.1 개체 검증 모델

개체 검증 모델의 목적은 군집 내의 개체들을 보고 해당 군집이 개체로써 인정받을 수 있는지를 평가하는 것이다. 먼저, 개체를 하나의 임베딩으로 나타낸다. 개체의 글자에 컨벌루션 신경망을 적용하여 단어 임베딩을 구한다. 다음으로 주변 단어, 주변 개체에 각각 LSTM을 적용하여, 주변 단어로 나타내어지는 문맥과 주변 개체로 나타내어지는 문맥을 구한다. 이렇게 얻은 단어 임베딩과 문맥 임베딩을 이어붙여 개체 임베딩을 나타낸다. 이후 군집 내의 개체 임베딩 각각에 2겹의 FFNN을 적용하여 해당 개체가 개체인지 아닌지를 구분하는 점수를 얻는다. 이 점수가 일정 이상이 되지 않으면 개체 임베딩에 0을 곱해 군집 내에 없는 것으로 취급한다. 이를 통해 군집에 개체만 남았다고 가정한다. 마지막으로, 군집 내의 개체 임베딩 전체에 CNN을 적용하여 군집에 대한 임베딩을 구한 뒤, 이를 2겹의 FFNN을 사용하여 군집에 대한 점수로 나타낸다. 이 점수가 군집이 등록될만한지를 판단하는 점수가 된다. 그림 3에 모듈의 흐름을 표시하였다.

3.5 개체 등록

마지막으로, 개체 등록에서는 검증된 개체 군집을 지식베이스에 등록하며 새로운 URI를 부여하는 작업이다. 지식베이스를 사용하는 다른 작업에 쓰기 위해서는 개체명과 개체 타입을 부여해야 개체로써의 모든 역할을 할 수 있지만, 본 연구에서는 개체명 및 개체 타입 부여는 향후 연구로 남겨놓고, 군집 내의 가장 많이 쓰인 개체명을 URI로 취급한다. 등록된 개체는 추후 다른 문서가 입력으로 들어올 시 등록된 개체를 연결할 수 있게 해야 한다. 이를 위해 새로 지식베이스에 등록될 URI는 개체명 사전, 개체 임베딩을 함께 가지고 있어야 한다. 본 연구에서는 개체 군집에서 등장한 표면형을 개체명 사전으로 등록하고, 개체 검증에서 사용한 임베딩의 평균값으로 등록한다.

3.6 개체 발견 과정 정리

3.1에서 3.5까지의 장치를 통해 말뭉치를 분석하면 기존의 지식베이스에 연결된 개체와, 검증되어 새로 지식베이스에 등록된 군집이 생긴다. 이 때, 새로운 텍스트가 들어올 경우, 지식베이스에 새로 추가된 개체 역시 개체 연결의 대상이 되어야 한다. 그림 2에서 각각의 모듈을 통과했을 때 텍스트에 어떤 것이 주석되는지를 표시해 놓았다.

4. 실험

4.1 데이터셋

본 논문에서의 지식베이스는 한국어 DBpedia 2016년 버전을 사용하였다.

영어 개체 연결 데이터로 자주 쓰이는 AIDA-CoNLL 데이터셋 [11]은 문장 내의 단어마다 해당 단어가 개체를 이루고 있는지, 해당 개체가 YAGO 지식베이스의 어떤 URI로 연결되어 있는지 등을 나타낸다. 해당 데이터셋에서는 특정 개체가

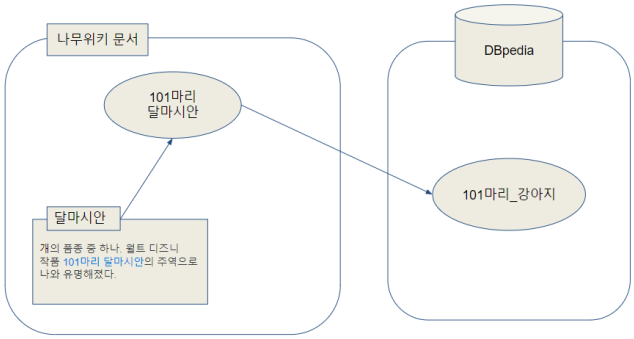

그림 4. 나무위키 데이터셋 수집 방법

타입	기존 개체	새 개체	총합
인명	229	189	418
지명	141	35	176
기관명	113	43	156
기타	121	108	229
합	604	375	979

표 1. 나무위키 데이터셋의 타입별 통계

YAGO 지식베이스 내에 존재하지 않는 경우 NIL 개체로 분류하였다. 하지만 본 논문에서 제시한 방법을 평가하기 위해서는 NIL 개체로 끝나는 것이 아니라, 새로운 개체를 주석해야 한다. 이를 위해 새로운 개체가 등장하는 말뭉치를 추출용 말뭉치와 테스트 말뭉치로 분리하여, 추출용 말뭉치에 등장한 새 개체를 테스트 말뭉치에서 얼마나 잘 뽑는지를 측정하는 방식을 사용한다.

본 연구에서 사용한 지식베이스는 한국어 DBpedia 2016년 버전이다. 이 때 추출 및 평가 말뭉치로 위키피디아를 사용할 경우, 지식베이스 내에 없는 개체를 찾기 어렵고, 지식베이스에 존재하지만 URI이 달라진 개체가 많아 사용이 부적절하다. 따라서 본 논문에서는 나무위키 문서 및 제목을 사용하여 데이터셋을 구축하였다. 나무위키 문서 제목을 URI로, 해당 문서로 링크되는 텍스트를 개체로 사용하여 많은 수의 데이터를 얻을 수 있다. 이 때, 나무위키 문서 제목을 DBpedia URI로 연결하는 과정이 필요한데, 이 과정은 인명, 지명, 기관명, 인공물 등의 카테고리를 가진 979개의 문서를 대상으로 전문가가 태깅하였다. 개체 카테고리에 대한 수집 통계는 표 1에 나타나 있다. 이후 이 979개의 개체를 나타내는 링크를 나무위키에서 수집하여 말뭉치로 사용하였다. 그림 4에 문서 및 말뭉치 수집 예시를 나타내었다.

나무위키 문서는 새로운 개체에 대한 발견의 지표를 나타내는 데 적합하지만, DBpedia 링크를 포함하고 있지 않기 때문에

	나무위키		위키피디아	
	추출	평가	추출	평가
문서 수	27809	3091	1468	1468
개체 수	264958	5318	88349	13910

표 2. 개체 수 통계

기존의 개체 연결 성능을 측정하는 데는 부족하다. 따라서, 개체 추가에 의한 기존의 개체 연결 성능의 변화를 측정하기 위해 한국어 위키피디아 문서를 대상으로 개체 연결을 수행하였다. 평가 말뭉치는 크라우드소싱을 통해 제작하였다. 나무위키 문서 및 위키피디아 문서와 링크 수집 수는 표 2에 나타나 있다. 나무위키와 위키피디아의 문서당 링크 수에 차이가 있는데, 이는 두 데이터셋을 모은 목적과 방법이 다르기 때문이다. 나무위키의 경우, 새로운 개체를 잘 찾는지를 확인하기 위해 모은 데이터셋이기 때문에 태깅된 개체를 나타내는 링크만을 수집하였다. 위키피디아의 경우는 기존의 개체 연결 성능을 평가하기 위해 모은 데이터이므로, 모든 링크 및 크라우드소싱을 통해 얻은 개체를 모두 사용하였다.

4.2 실험 및 평가 방법

반복적 개체 연결의 한 바퀴가 종료되면, 지식베이스에는 검증된 URI가 추가되고, 개체 연결 모듈에는 갱신된 개체명 사전과 새로운 개체 임베딩이 추가된다.

개체 검증과 반복적 개체 연결에 대한 평가는 올바른 개체를 등록하였는지, 새로운 개체가 잘 연결되는지, 새로운 개체가 등록되고 연결되는 것이 기존의 개체 연결 성능에 도움이 되는지에 대한 기준으로 측정해야 한다.

실험은 다음과 같이 진행된다. 먼저, 말뭉치를 발견 말뭉치와 평가 말뭉치로 9:1의 비율로 나눈다. 발견 말뭉치에서는 위에서 제시한 반복적 개체 연결의 한 바퀴를 수행한다. 이 때 발견 말뭉치 내의 새로운 개체들이 등록된다. 군집 내에서 다수를 차지하는 개체가 나타내는 URI를 개체의 URI로 등록한다. 마지막으로, 새로운 개체가 등록된 지식베이스를 기반으로 평가 말뭉치에서의 개체 연결 성능을 측정한다. 개체 연결에서의 정밀도는 시스템이 지식베이스 내에 연결한 개체 중 올바르게 연결한 URI의 비율로 측정하고, 재현도는 정답 개체 중 지식베이스 내에 연결된 URI를 대상으로 시스템이 올바르게 맞춘 것의 비율로 측정한다 [12].

4.2.1 가짜 개체 군집

본 논문에서는 개체 연결 중 개체명 사전에 없어서 개체 연결을 하지 못한 경우를 상정하지 않기 때문에, 개체 군집에 대해 다수의 개체가 나타내는 URI를 등록하는 방식으로 성능을 측정하였다. 하지만 이 방법은 단순히 개체명을 많이 등록할수록

	나무위키			위키피디아		
	F1	Fake	Δ Link	F1	Fake	Δ Link
모든 개체 미등록	70.91	0	0	76.31	0	0
모든 개체 등록	81.34	2966	841	73.52	27	13
검증 모듈	78.77	1828	638	75.80	40	12
No-Fake	82.30	0	790	74.14	0	14

표 3. 실험 결과. Fake는 가짜 군집에 연결한 개수이고, ΔLink는 모든 개체를 미등록한 경우 대비 새로 연결한 개체의 수이다.

	나무위키	위키피디아
미등록	1205	3336
등록 불가능	172	2549

표 4. 평가 말뭉치에서 개체명 사전에 등록되지 않은 개체명의 수 및 등록 불가능한 개체명의 수. 미등록은 평가 말뭉치에서 기존의 개체명 사전에 등록되지 않은 개체명의 수를 의미한다. 등록 불가능은 발견 말뭉치에서 모든 개체명을 등록하더라도 개체명 사전에 등록되지 않는 개체명의 수를 의미한다.

성능이 오르게 된다. 따라서, 이에 대해 보정하기 위해 같은 표면형을 가지지만 다양한 URI를 가리키는 개체들을 모아 가짜 군집을 만들었다. 이 방법으로 단순히 모든 군집을 등록하는 것에 페널티를 주었다. 또한, 가짜 개체 군집은 기존의 개체 연결 성능을 평가하는 지표로도 활용이 가능하다. 기존의 개체명 또한 가짜 군집으로 등록되기 때문에, 매핑 사전의 확률값이 변화하여 후보 생성 과정에 영향을 주게 된다.

4.3 Baseline

본 연구에서의 baseline으로는 검증 없이 모든 군집을 등록하는 모델과, 모든 개체를 걸러내는 모델을 비교하였다. 모든 개체를 걸러내는 모델은 지식베이스에 새로운 URI가 등록되지 않으므로, 지식베이스의 증감에 따른 개체 연결의 범위를 측정할 수 있다. 모든 개체를 등록하는 모델은 오류 보정 없이 모든 개체를 등록하므로 각종 노이즈를 포함하게 될 것이다. 마지막으로, 가짜 개체 군집을 완벽하게 배제하는 No-Fake를 넣어 개체 발견으로 얻을 수 있는 최대한의 성능과 비교하였다.

본 논문에서 개체명 인식 모듈은 ETRI에서 만든 언어분석기를 사용하였다. 개체 연결 모듈으로는 [4]의 모델을, 개체 군집화 모듈으로는 [10]의 모듈을 사용하였다.

5. 결과 및 분석

5.1 실험 결과

전체 실험 결과는 표 3에 나타나 있다.

나무위키 데이터셋에서는 모든 개체를 등록하는 경우가 개체 검증 모듈을 사용했을 때보다 F1점수가 더 좋은 것을 확인할 수 있다. 하지만 위키피디아의 경우, 개체를 등록하지 않는 경우가 개체를 등록하는 경우보다 성능이 좋음을 알 수 있다. 개체 검증 모듈을 사용한 경우, 두 데이터셋에서 모두 평균 이상의 성능을 보였다.

5.2 개체 발견의 영향

나무위키 데이터셋에서는 나무위키 링크만을 평가 대상으로 삼기 때문에, 지식베이스에 없던 URI를 많이 등록할수록 성능이 좋아지게 된다. 모든 개체를 등록할 때 가장 많은 개체를 발견하게 되고, 이에 따라 개체 연결의 성능이 크게 올랐음을 알 수 있다. 개체 검증 모듈을 사용한 경우 역시 모든 개체를 등록했을 때만큼은 아니지만 개체를 아예 등록하지 않았을 경우보다 성능이 크게 오른 것을 알 수 있다. No-Fake의 경우, 나무위키 데이터셋에서 모든 개체를 등록했을 때보다 1점이 더 오른 것을 알 수 있다. 이는 가짜 개체 군집이 개체 발견에도 부정적인 영향을 미친다는 것을 의미한다.

5.3 개체 검증의 영향

위키피디아 데이터셋에서 모든 개체 군집을 등록하지 않은 경우와 타 모듈을 비교하면 개체 등록이 기존의 개체 연결 모듈에 얼마나 영향을 미쳤는지 알 수 있다. 모든 개체를 등록한 경우 새로운 개체명을 다수 등록할 수 있다. 이는 새로운 개체를 발견하는 것이 목적인 나무위키 데이터셋에서 높은 성능을 보이는 이유이다. 하지만 위키피디아 데이터셋에서의 점수를 보면, 개체를 등록하지 않았을 때보다 3점 가까이 점수가 하락한 것을 알 수 있다. 이는 불필요한 개체명을 다수 등록하여 기존에 있던 개체명 사전에 악영향을 미쳐 후보 생성 과정에서 노이즈가 생겼다고 볼 수 있다. 또한, 가짜 개체를 모두 배제하는 No-Fake의 성능 역시 개체 미등록에 비해서는 2점, 검증 모듈을 사용했을 때에 비해 1점 이상 떨어졌다. 그 이유는 가짜 군집을 배제하더라도 군집화 모듈에서 하나의 개체를 나타내는 군집화를 잘 하지 못했다는 의미로 해석할 수 있다.

5.4 개체의 표면형

개체 등록을 통해 등록된 표면형의 영향을 알아보기 위해 발견 말뭉치와 평가 말뭉치간의 표면형을 분석하였다. 표 4에서 미등록 개체가 많을수록 개체 발견을 하지 않았을 때의 성능이 낮을 것이다. 등록 불가능한 개체명의 수가 많다면, 개체 발견을 하더라도 성능이 크게 늘지 않을 것이다.

나무위키 데이터셋의 경우, 개체 발견을 수행한다면 미등록 개체명 중 대다수가 등록되고, 이는 차후 개체 연결에 사용되어 개체 연결의 범위가 효과적으로 늘어나게 된다. 그러나 위키피디아의 경우는 미등록 개체명 중 등록 불가능한 개체명의 비율이 높고, 전체 개체 중에서도 그 비율이 높은 것을 알 수 있다. 이 경우 개체 발견이 큰 영향을 주지 않고, 오히려 잘못된 개체 군집을 등록할 가능성이 높아 성능이 하락하는 원인이 된다.

6. 결론

본 논문에서는 지식베이스의 자동 확장을 위한 개체 발견 과정을 정의하고, 새로 지식베이스에 등록된 URI들이 개체 연결에 어떤 영향을 미치는지 실험하였다. 실험 결과, 새로운 개체를 발견하는 데는 모든 표면형을 등록하는 것이 좋은 성능을 보이지만, 이와 같이 무분별하게 등록하는 경우 기존의 개체 연결 성능에 악영향을 미치기 때문에 적절한 검증 모듈이 필요하다는 것을 보였다.

이후 연구에서는 개체명 사전에 없어서 생긴 문제를 함께 고려하고, 개체명 사전에 없어서 연결에 실패한 경우를 해결하며 동시에 개체를 발견하는 연구를 할 예정이다. 또한, 군집 내에서 노이즈가 될만한 개체를 등록 전에 걸러내는 방식 등으로 노이즈를 제거하는 연구가 필요하다.

감사의 글

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2013-0-00109, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발)

참고문헌

[1] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 708–716, 2007.

[2] N. Gupta, S. Singh, and D. Roth, "Entity linking via joint encoding of types, descriptions, and context," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2681–2690, Sep. 2017. [Online]. Available: https://www.aclweb.org/anthology/D17-1284

[3] O. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," *CoRR*, Vol. abs/1704.04920, 2017. [Online]. Available: http://arxiv.org/abs/1704.04920

[4] P. Le and I. Titov, "Improving entity linking by modeling latent relations between mentions," *arXiv preprint arXiv:1804.10637*, 2018.

[5] S. Tamang, Z. Chen, and H. Ji, "Cuny blender tackbp2012 entity linking system and slot filling validation system." *TAC*, 2012.

[6] J. Hoffart, Y. Altun, and G. Weikum, "Discovering emerging entities with ambiguous names," *Proceedings of the 23rd international conference on World wide web*, pp. 385–396, 2014.

[7] Z. Wu, Y. Song, and C. L. Giles, "Exploring multiple feature spaces for novel entity discovery," *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[8] M. Van Erp, F. Ilievski, M. Rospocher, and P. Vossen, "Missing mr. brown and buying an abraham lincoln-dark entities and dbpedia." *NLP-DBPEDIA@ ISWC*, pp. 81–86, 2015.

[9] P. Vossen, R. Agerri, I. Aldabe, A. Cybulska, M. van Erp, A. Fokkens, E. Laparra, A.-L. Minard, A. P. Aprosio, G. Rigau, M. Rospocher, and R. Segers, "Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news," *Special Issue Knowledge-Based Systems, Elsevier*, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705116302271

[10] J. Shen, R. Lyu, X. Ren, M. Vanni, B. Sadler, and J. Han, "Mining entity synonyms with efficient neural set generation," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 249–256, 2019.

[11] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 782–792, 2011.

[12] W. Shen, J. Wang, and J. Han, "Entity linking with

a knowledge base: Issues, techniques, and solutions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, No. 2, pp. 443–460, 2014.