인지과학, 제26권 제4호
Korean Journal of Cognitive Science
2015, Vol. 26, No. 4, 435～452.

유저 모델과 실시간 뉴스 스트림을 사용한 트윗 개체 링킹*

정 소 윤　　박 영 민　　강 상 우†　　서 정 연
서강대학교 컴퓨터공학과

최근 개체 링킹에 대한 연구들은 지식 베이스를 외부 자원으로 사용하여 실세계의 지식과 의미적인 관련도를 통해 중의성을 해소하는데 중점을 두고 있다. 지식 베이스를 사용한 개체 링킹은 신문기사나 블로그 포스트 등에서는 좋은 성능을 보이지만, 마이크로블로그에서는 짧은 텍스트 길이와 지식 베이스에 존재하지 않는 주제를 다루는 특성 때문에 비교적 낮은 성능을 보인다. 본 논문에서는 140자가 되지 않는 짧은 텍스트 내에서 실시간으로 빠르게 정보를 공유하는 특성을 가지는 마이크로블로그에서 나타나는 개체명의 중의성을 해소하는 방법을 제안한다. 제안하는 방법은 지식 베이스만 사용하는 개체 링킹의 한계를 극복하기 위해 마이크로블로그 사용자 기록과 뉴스 기사를 이용하고, 지식 베이스에 존재하는 특정 엔트리로 개체 링킹을 수행한다. 본 논문에서는 개체명을 포함하는 한국어 트윗을 추출하여 데이터를 구축하였다. 성능 평가는 정확도 지표(시스템이 정답으로 판정한 데이터 개수/전체 데이터 개수)를 사용하였으며, 제안하는 시스템은 구축한 데이터에서 기존 지식 베이스만 사용한 개체 링킹 시스템보다 높은 67.7%의 정확도를 나타내었다.

주제어 : 개체중의성해소, 개체링킹, 트윗, 위키피디아

---

* 이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.NRF-2013R1A1A2010190).

† 교신저자: 강상우, 서강대학교 컴퓨터공학과, 서울특별시 마포구 백범로 35 R904
연구분야: 자연어처리
E-mail: gahng.sw@gmail.com

서 론

최근 인터넷과 컴퓨팅 기술의 발전, 모바일 기기와 센서들의 진화, 네트워크의 출현 등으로 정보량이 급속도로 늘어나고 있다. 따라서 증가하는 정보들 가운데 필요한 정보를 찾기 위한 다양한 연구들이 진행되고 있다. 정보 추출의 한 분야인 개체명 인식과, 인식된 개체명을 특정 개체에 링킹하는 연구들은 방대한 정보 속에서 의미 있는 지식을 추출하기 위해 활발히 시도되고 있다. 개체 링킹(Entity Linking)은 텍스트에 출현한 개체명을 위키피디아와 같은 지식 베이스의 특정 엔트리[1]에 대응시키는 작업이다.

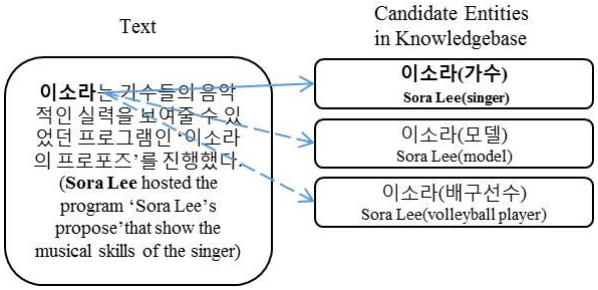

그림 1. 개체 링킹의 예

그림 1에서 왼쪽 텍스트 상자의 굵은 글씨가 중의성을 가지는 개체명 "이소라"이고, 실선으로 이루어진 화살표가 가리키는 "이소라 (가수)"가 링킹이 되어야 하는 지식 베이스의 정답 개체명이다. 중의성 해소를 위해 대상 개체명 인접 문맥 정보와 지식 베이스를 외부 자원으로 활용하는데, 개체명 인접 문맥 정보의 형태는 지식 베이스의 엔트리가 될 수 있다. 예를 들어, 왼쪽 텍스트 상자에서는 "이소라의 프로포즈"가 지식 베이스의 엔트리에 존재하는 개체이고, 지식 베이스에서

1) 본 논문에서 개체와 동일한 의미로 사용되며 각 페이지의 텍스트가 설명하는 대상을 나타낸다. 또한 페이지의 텍스트 내에는 의미적 관계를 가지는 다른 개체들의 페이지들이 링크로 나타난다. 다시 말해, 위키피디아 개체들 사이에 링크가 존재하면, 개체들은 의미적 관계를 갖는다.

“이소라 (가수)”가 진행한 프로그램이라는 정보를 제공해 주므로, 올바르게 중의성 해소가 될 수 있다.

최근에는 신문 기사와 같은 정형화된 텍스트 내에서의 개체 링킹 뿐 만 아니라, 짧고 비정형적인 텍스트 내에서의 개체 링킹에 대한 연구가 해외에서 활발히 진행되고 있다. 예를 들면 대표적인 마이크로블로그 서비스 중 트윗에서의 연구가 활발하다. 트윗은 사용자들이 실시간으로 140자 이내의 트윗이라는 글을 쓰는 행위를 통해 의사소통하는 공간이며 트윗은 개인의 관심 분야나 일상생활, 사회적 이슈 등의 주제에 대해 실시간으로 빠르게 전파되는 특성을 지닌다(Java, A. et al., 2007). 트윗에서의 개체 링킹은 구조적 지식 베이스인 위키피디아와 같은 곳에서 실세계 지식을 통해 해소되어 왔다. 트윗의 특성을 고려하였을 때, 최근의 연구들에서는 사용자 개인의 관심 분야 모델링을 통해 개체 링킹을 수행하기도 하였다 (Shen, W. et al., 2013, Bansal, R. et al., 2014). 본 논문에서는 실시간으로 일어나는 사건들에 대한 소통이 많이 이루어지는 트윗의 특성을 고려하여 기존 연구에서 사용되지 않았던 뉴스 기사를 외부 자원으로 사용함으로써 트윗 개체 링킹의 성능을 향상시킨다.

본 논문에서는 지식 베이스 뿐 만 아니라, 이슈 모델링과 유저 모델링을 통해 트윗에서의 개체 링킹을 제안한다. 본 논문의 구성은 다음과 같다. 관련 연구에서는 개체 링킹에 관련된 기존 연구에 대해 설명하며 다음 장에서는 뉴스기사와 유저 모델링을 통한 트윗 개체 링킹 시스템에 대해서 설명하고, 자체적으로 구축한 한국어 트윗 코퍼스와 제안하는 시스템의 실험 및 결과를 제시한다. 마지막으로는 연구의 요약과 결론을 제시한다.

관련 연구

개체 링킹을 위해서는 대상으로 하는 개체들을 파악해야 하므로 필수적으로 개체명 인식이 우선하여야한다. 지식 베이스 개체명 인식 문제는 전통적인 개체명 인식과는 차이가 있다. 전통적인 개체명 인식에서는 개체에 장소나 단체 등의 클래스를 정해주는 반면, 지식 베이스 기반 개체 링킹에서의 개체명 인식은 텍스트

내 모든 가능한 지식베이스에 존재하는 개체들의 정규화 된 명칭들의 후보들을 추출하는 것이다. 예를 들어, '이소라'라는 개체에 일반적인 개체명 인식에서는 "PER"등의 클래스로 분류하지만, 지식 베이스 개체명 인식에서는 "이소라 (가수)", "이소라 (모델)" 등의 개체명으로 태깅해야 한다. 개체 링킹의 초기 모델들은 텍스트 문서 내 모든 가능한 n-gram 용어들 중 개체명 사전에 해당하는 것들만 추출하는 방식이 시도되었다(Mihalcea, R., & Csomai, A., 2007). 최근에는 개체명이 정답으로 부여된 학습 문서를 분류기로 학습하여 개체명을 인식하는 방법이 시도되었다(Milne, D., & Witten, I. H., 2008). 한국어 개체 링킹 연구에서는 각 개체명이 나타날 수 있는 surface form을 미리 사전으로 구축하여, SVM을 이용한 개체경계 인식 방법을 제시 한 바 있다(김영식 외, 2014). 트윗에서의 개체명 인식은 영문 트윗을 대상으로 KNN classifier와 CRF labeler를 하이브리드 방식으로 사용하여 반지도 방식으로 시도된 바 있다(Liu, X. et al., 2011).

개체명 인식 후에는 개체 링킹의 마지막 단계인 인식된 개체명 중 중의성을 가지는 개체명의 중의성 해소단계를 거친다. 기존 연구에서는 중의성 해소를 위해 공기 개체 의미 관련도가 활용되고 있다. Bunescu, R. C.와 Pasca, M.(2006)의 연구는 처음으로 위키피디아 카테고리 정보를 가지고 의미 관련도를 이용한 유사도 측정법을 정의하였다. 중의성을 가지는 개체명 주변 용어들과 위키피디아 문서 내에 나타나는 용어들의 tf-idf cosine similarity를 이용하여 개체 의미 관련도를 제안하고, 위키피디아 카테고리 정보를 사용하여 성능을 개선했다. Cucerzan, S.(2007)은 Bunescu, R. C.와 Pasca, M.(2006)와 같은 과정을 거치지만, 위키피디아 문서 내 혹은 중의성을 가지는 개체를 포함하는 문맥 내 모든 용어를 사용하지 않고, 위키피디아에서 제공하는 개체명 표현만을 사용하였다. Milne, D.와 Witten, I. H.(2008)은 공기 개체 의미 관련도를 중의성을 가지는 개체의 인접한 비중의성 공기 개체들을 사용하여 평균 의미관련도와 개체 선험 확률을 위키피디아 말뭉치로 학습시킨 분류기를 통해 개체 링킹을 수행하였다. Han, X.와 Zhao, J.(2009)는 Pagerank 알고리즘을 통해 집단적 개체중의성해소를 시도하였다. 중의성을 가지는 개체명을 포함하는 문서 내 각 개체명에 노드들과 개체명의 모든 가능한 지식 베이스 내 개체명에 대응하는 개체 노드들을 생성하였고, 노드 간 edge 가중치로 개체명 노드와 지식 베이스 내 개체명 노드 사이에 지역 문맥 유사도를 설정하고, 개체노드 간에 개체

쌍의 의미 관련도를 계산하였다. 또한 개체명이 지식 베이스 내 개체명과 동일하게 나타나지 않는 개체명 표현을 구축하여 개체명 인식 오류를 줄여 개체 중의성 해소의 성능을 높인 연구가 최근 발표되었다(Charton, E. et al., 2014).

트위터에서의 개체 링킹은 마이크로블로그에서의 정보 추출이 주목받으며 많은 연구들이 이루어져 왔으며 Derczynski, L. et al.(2015)은 새로운 트위터 데이터를 구축하고 최신의 방법들을 구현하여 비교하였다. 트위터 개체 링킹 연구들 중 유저 모델링을 사용하는 연구는 노이즈가 많은 텍스트를 다루기 때문에 불충분한 문맥 정보라는 한계점을 극복하기 위해 유저 모델링을 통한 개체 링킹 연구들이 진행되고 있으며 최근 그래프 기반의 KAURI(Knowledge bAse via UseR Interest modeling) 시스템이 제안되었다(Shen, W. et al., 2013). KAURI는 사용자의 이전 모든 트윗에서 나타난 모든 개체명들을 사용자 토픽 모델링을 통해 그래프를 구축하여 중의성을 해소한다. 그래프의 노드 집합으로, 과거 사용자의 모든 트윗으로부터 모든 개체명에 대응하는 개체표현 노드들과 개체명의 모든 가능한 지식 베이스에 존재하는 후보 개체에 대응하는 개체 노드들을 생성하였다. 노드 간 엣지 가중치로는 개체표현 노드와 개체 노드 사이에 지역 문맥 유사도를 설정하고, 개체노드 간에 의미 관련도를 설정하였다. 또 다른 연구에서 사용자의 이전 트윗의 분석을 통한 유저 모델과 문맥 모델(contextual model)의 하이브리드 방식으로 개체명의 중의성을 해소하는 방법을 제안되었다(Bansal, R. et al.., 2014). 유저 모델링을 통한 개체 링킹은 단순 지식 베이스만 사용한 성능보다 트윗 개체 링킹에서 더 좋은 성능을 나타내었다.

본 연구에서는 한국어 트윗에 대한 개체 링킹 방법을 개체명 인식 단계에서는 전통적인 n-gram 용어들 중 개체명 사전에 해당하는 것들만 추출하는 방식을 사용하였고, 개체 중의성 해소 단계에서 트윗의 특성에 맞는 3가지 모델들을 제안함으로써 트윗 개체명 중의성 해소에 초점을 맞추고 있다.

트윗 개체 링킹 모델

본 논문에서 제안하는 시스템은 트윗 사용자의 관심 분야와 실시간 사회적 이

슈, 그리고 트윗 발언 내의 문맥적 정보를 고려하는 방법을 적용한다. 제안하는 시스템은 그림 2와 같이 세 가지 모델로 구성된다. 문맥 모델(Contextual model), 사용자 모델(User model) 그리고 이슈 모델(Issue model)로 이루어지고 링킹 모델(Linking model)이 이들을 통합한다. 문맥 모델에는 기존의 방법을 적용하였는데, 중의성을 가지는 개체명 주변 문맥의 비중의성 공기 개체들만을 사용하여 중의성을 해소한다. 사용자 모델은 중의성을 해소하고자 하는 개체명을 포함한 트윗을 게시한 사용자의 모든 트윗 기록들을 수집하여 중의성을 해소한다. 이슈 모델은 위키피디아에서 다루지 않는 자질을 외부 자원인 뉴스 기사부터 추출한다. 링킹 모델은 세 모델의 스코어를 통합하여 가장 점수가 높은 개체 후보로 개체명을 링킹한다.

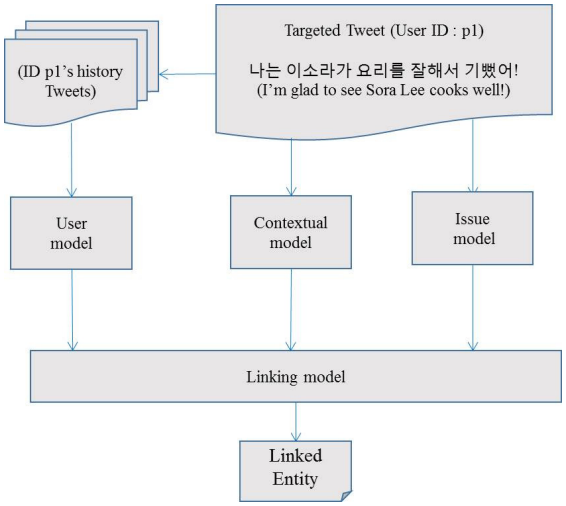

그림 2. 제안하는 개체 링킹 모델

프레임 워크

- E - 중의성이 해소되어야 하는 개체명
- e^j - 위키피디아 엔트리에 존재하는 j번 째 비중의성 개체
- c_j - E가 링크 될 수 있는 위키피디아 엔트리에 존재하는 j번 째 후보 개체

- $\langle D \rangle$ - 문서 D에 나타나는 모든 개체 집합
- $D_{c_j}^i$ - c_j를 주제로 하는 i번 째 뉴스 기사, D_{c_j}는 모든 $D_{c_j}^i$의 집합
- $[e]$ - 위키피디아 엔트리 e가 가지는 위키피디아 링크의 집합
- $S_c(c_j), S_u(c_j), S_i(c_j)$ - j번 째 후보 개체의 문맥 모델, 유저 모델, 이슈 모델의 스코어

문맥 모델

문맥 모델은 문맥 정보를 사용한다. 여기서 문맥 정보란, 중의성을 가지는 개체명의 인접한 비중의성 개체들을 이야기한다. 하나의 트윗에 중의성을 가지는 개체명이 존재 할 경우 트윗에 포함된 모든 비중의성 개체들이 해당된다. 대부분의 위키피디아를 외부자원으로 사용한 기존 연구에서 사용된 의미 관련도는 인접 용어 집단 뿐 아니라 위키피디아 카테고리 정보도 사용하였다. 하지만 위키피디아 카테고리 정보를 사용한 경우 사용하지 않은 경우보다 성능이 낮게 나오는 연구 결과가 존재한다.(Milne, D & Witten, I. H., 2008). 위키피디아는 사용자가 그리고 한국어 위키피디아는 영어 위키피디아 보다 비교적 아직 카테고리 정보가 불충분하기 때문에 본 연구에서는 카테고리 정보를 사용하지 않았다. 문맥 모델의 스코어링 방법은 Charton, E. et al..(2014)의 mutual relation score"를 $S_c(c_j)$로 나타내고, 이를 전체 시스템의 베이스라인으로 사용한다(식 (1)~(3)). $S_c(c_j)$은 $dsr_{score}(e^i, c_j)$ (식 (1))와 $csr_{score}(e^i, c_j)$ (식 (2))의 가중 합으로 이루어진다. $dsr_{score}(e^i, c_j)$ 은 중의성 개체가 나타나는 문서에 존재하는 i번 째 비중의성 개체 e^i가 후보 개체 c_j의 위키피디아 페이지에 링크로 출현하는 횟수를 계산한다. $csr_{score}(e^i, c_j)$ 은 중의성 개체가 나타나는 문서에 존재하는 i번 째 비중의성 개체 e^i의 위키피디아 페이지에 나타나는 링크와 후보 개체 c_j의 위키피디아 페이지의 링크 정보가 겹치는 횟수를 계산하는 수식이다.

$$S_c(c_j) = \sum_{e^i \in \langle D \rangle} \delta\, dsr_{score}(e^i, c_j) + (1-\delta)\, csr_{score}(e^i, c_j) \tag{1}$$

$$dsr_{score}(e^i, c_j) = \left| e^i \cap [c_j] \right| \tag{2}$$

$$csr_{score}(e^i, c_j) = \frac{\left| [e^i] \cap [c_j] \right|}{|[e^i]| + |[c_j]|} \tag{3}$$

그림 3. 중의성을 가지는 개체인 “이소라”의 위키피디아 페이지

위키피디아에는 동명이인 문서 “disambiguation page”가 있다. 그림 3에서 확인할 수 있듯이 “이소라” 동명이인 문서에는 세 명의 후보를 가진다. 첫 번째 “이소라”는 대한민국의 모델이고, 두 번째는 대한민국의 가수이다. 마지막 “이소라” 는 대한민국 배구 선수이다. 예를 들어, E가 “이소라”라면, 위 식 (1)에서 계산된 $S_c(c_j)$가 각각 “이소라 (가수)”, “이소라 (모델)”, “이소라 (배구선수)”에 대해 계산되고, 이때, e^i는 중의성 개체가 발견되 트윗에서 나타난 비중의성 개체가 된다. $S_c(c_j)$가 가장 높은 점수를 가지는 후보 엔트리로 링킹된다.

유저 모델

트윗은 사용자의 관심 분야와 실시간 사회적 사건에 대해 의사소통이 이루어진다는 특성이 있다. 유저 모델은 이러한 특성을 이해하여 사용자의 행동과 관심분야를 다루기 위해 사용자의 이전 트윗 기록을 모두 사용한다. 이 때, 사용자의 이전 트윗에서 나타나는 개체명들에 대해 사용자는 관심과 흥미를 가지고 있다고 가정한다.

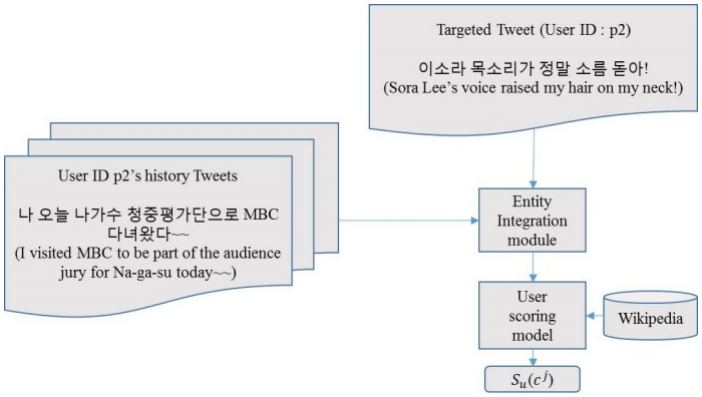

그림 4. 유저 모델

$$S_u(c_j) = \sum_{e^i \in \langle D \rangle} \delta\, dsr_{score}(e^i, c_j) + (1-\delta)\, csr_{score}(e^i, c_j) \tag{4}$$

$$\langle D \rangle = \{e^i | D = \text{사용자의 과거 트윗 기록부터 현재 트윗까지의 텍스트}\}$$

그림 4는 유저 모델의 구조를 보여준다. 유저 모델이 중의성을 가지는 개체명을 특정 트윗에서 인식하면, 그 트윗을 남긴 사용자의 이전 모든 트윗을 추출한다. 개체 통합 모듈(Entity Integration module)은 좌최장일치방법과 어절 uni-gram과 bi-gram 자질을 이용하여 미리 구축한 위키피디아 엔트리 사전에 해당하는 개체가 존재하면 추출하여 $< D >$에 추가한다. D의 위키피디아 엔트리들은 대부분 명사구로 이루어져 있다. 하지만 트윗 데이터의 특성상 노이즈가 많아 형태소 분석의 성능이

떨어지기 때문에 개체 추출 시 좌최장일치법을 사용하고 자질로는 어절 uni-gram과 bi-gram을 사용한다(Kang, S. et al., 2014). 최종적으로 식(4)에 의해 유저 모델의 스코어가 산출된다. 그림 4의 예제에서, 시스템은 중의성을 가지는 개체인 “이소라”를 사용자 p2의 트윗에서 발견하고, p2의 모든 과거 트윗 기록을 수집한다. 이 때 수집된 모든 과거 트윗 기록은 D라고 볼 수 있다. 개체 통합 모듈은 과거 p2의 트윗 기록에서 “이소라 (가수)”가 출현했던 “나가수”와 방송사인 “MBC” 와 같은 개체명인 e^i를 추출한다. 이 자질들은 링크 후보 개체인 c_j중 “이소라 (가수)”가 최종적으로 링킹 될 확률을 높여주는데, 왜냐하면 “이소라 (가수)”의 위키피디아 페이지에는 “나가수”와 “MBC”라는 개체명을 링크로 가지고 있기 때문에 S_u(이소라(가수)) 의 점수가 높아지기 때문이다.

이슈 모델

이슈 모델은 문맥 모델과 유저 모델이 다루지 못하는 실시간 사회적 사건들이나 사소한 대중들이 관심을 가지는 사건들은 다룬다. 예를 들면, 유명 연예인의 실시간으로 일어난 사건이나 아주 사소한 사건으로 위키피디아에서는 다루지 못하지만 트윗에서는 많이 다루어지는 주제들을 뜻한다. 뉴스 기사는 위에서 말한 사건들을 다루는 특성을 지니고 있으므로, 이슈 모델은 뉴스 기사를 외부 자원으로 사용한다.

그림 5를 보면, 이슈 모델이 중의성을 가지는 개체명인 “이소라”를 이슈 모델과 같은 방법으로 인식하면, 뉴스 링킹 모듈(News linking module)이 인식된 트윗 게시 날짜의 k일 전 후로 “이소라”가 제목에 포함 된 뉴스 기사를 수집한다. 수집 된 뉴스 기사들은 제목에 나타난 “이소라”의 동명이인 페이지에 나타난 위키피디아 엔트리들 “이소라 (가수)”, “이소라 (모델)”, “이소라 (배구선수)”의 페이지에 나타난 텍스트와 코사인 유사도를 통해 특정 후보 엔트리 c_j와 링킹 된다. 위 예제에서는 후보 엔트리가 c_1, c_2, c_3가 각각 “이소라 (가수)”, “이소라 (모델)” 그리고 “이소라 (배구선수)”가 된다. 다시 말해 이슈 모델은 뉴스 기사 각각 특정 위키피디아 엔트리와 링크함으로써 뉴스 기사 하나를 하나의 위키피디아 페이지로 취급한다. 각

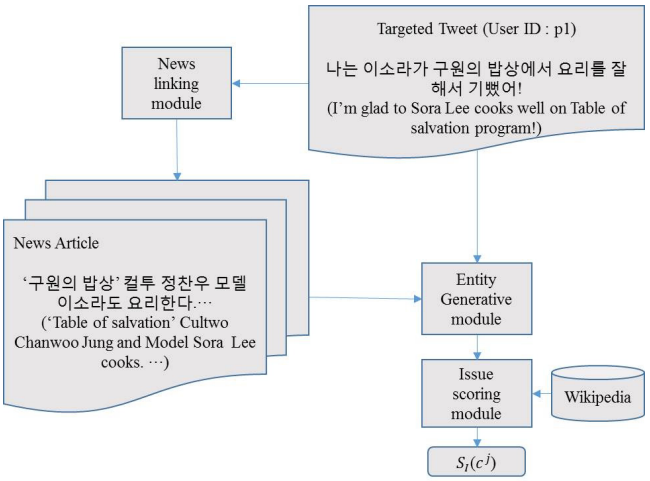

그림 5. 이슈 모델

기사는 후보 엔트리 c_j를 주제로 하는 i번 째 뉴스 기사 $D_{c_j}^i$가 된다. 개체 생성 모듈(Entity Generative module)은 각 뉴스 기사마다 위키피디아 링크를 생성한다. 신문 기사 속 작은따옴표는 이름 표시 기능을 가지는데, 요컨대 이름 표시를 할 때는 신문사마다 약간의 차이가 있으나, 책 이름, 영화 이름, 음반 이름, 드라마 이름 등이 작은따옴표로 묶일 수 있다(이동혁, 2008). 그러므로 개체 생성 모듈은 뉴스 기사의 작은따옴표로 명시된 용어들을 의미 있는 개체명이라 가정하고, 모두 링크로 생성한다. 개체 생성 모듈은 링크 생성 규칙에 따라 각 기사 마다 링크를 생성하고, 표 1은 개체 생성규칙과 그림 5의 예제에서 개체가 생성된 예제를 보여준다.

표 1. 이슈 모델에서 개체 생성 모듈의 링크 생성 규칙

개체 생성 규칙	생성 예제
작은따옴표 내부 용어	"구원의 밥상"
위키피디아 엔트리에 존재하는 개체명	"컬투", "정찬우", "모델"

$$S_i(c_j) = \frac{\sum_{i=0}^{|D_{c_j}|} \sum_{e^k \in \langle D \rangle} \delta\, dsr_{score}(e^k, D_{c_j}^i) + (1-\delta)\, csr_{score}(e^k, D_{c_j}^i)}{|D_{c_j}^i|} \quad (5)$$

표 1은 개체 생성 모듈의 링크 생성 규칙과 그림 5의 예제에 링크가 어떻게 생성되는지 보여준다. “구원의 밥상”은 위키피디아에는 실제로 없는 엔트리이지만, “이소라 (모델)”이 새로 진행하게 된 프로그램의 제목으로, 중의성을 가지는 개체명 “이소라”가 c_2인 “이소라 (모델)”로 링킹 될 수 있는 중요한 개체 정보가 될 수 있다. 두 번째 생성 규칙은 뉴스 기사를 형태소 분석하여 명사 자질만 사용하여, 명사 uni-gram, bi-gram이 위키피디아 개체명 사전에 존재 할 경우 추가한다. 예제에서는 “컬투”, “정찬우”, “모델”이 해당한다. 최종적으로 이슈 모델은 식 (5)과 같이 스코어링 한다. 표 1의 예제를 적용해 본다면, c_j가 “이소라 (모델)”일 때, $D_{c_j}^i$는 “이소라 (모델)”을 제목에 포함하는 i번째 뉴스 기사이고, 개체 생성 모듈의 링크 생성 규칙에 의해 뉴스 기사에 링크를 생성하여 트윗을 남긴 사용자의 기록을 모은 트윗 문서 D에 나타나는 비중의성 개체들의 위키피디아 페이지에 나타나는 링크 정보의 동시 출현 횟수를 점수에 반영한다.

링킹 모델

링킹 모델은 최종적으로 위 3가지 모델의 스코어를 통합하여 “Total Relatedness”를 계산한다. 최종 스코어는 식 (6)에서 $S_c(c_j), S_u(c_j), S_i(c_j)$는 가중치를 반영한 합으로 계산된다. 가중치 매개변수 α, β, γ는 트위터 사용자들이 문맥 모델, 유저 모델 그리고 이슈모델이 고려하는 사용자의 흥미를 트윗에 반영하는 정도, 실시간 이슈를 트윗에 반영하는 정도를 나타내고 이는 실험적으로 실험 성능이 가장 높을 때의 값으로 정하여 추정하였다.

$$TR(E, c_j) = \alpha S_c(c_j) + \beta S_u(c_j) + \gamma S_i(c_j) \quad (6)$$
$$(\alpha + \beta + \gamma = 1)$$

실험 및 평가

본 논문에서 실험 대상은 위키피디아 동명이인 문서 내에서 중의성을 갖는 사람 개체명이며 실험을 위하여 위키피디아 카테고리 정보를 이용하여 동명이인 개체명 사전을 구축하였다.

실험 데이터 구축을 위하여 최근 트윗을 활발히 이용하는 300명의 트윗 사용자 당 사용자 당 50~60개씩, 총 16367개의 트윗을 수집하였다. 수집된 트윗들 중 동명이인 개체명 사전에 존재하는 개체명이 포함된 트윗들을 선별하였다[2]. 선별 방법으로는 신뢰성 검증을 위해 3명의 실험자가 동명이인 중 정답을 태깅할 수 있는 트윗을 대상으로 하였으며 총 248개의 한국어 트윗 데이터를 구축하였다. 수집된 248개의 트윗 데이터 내에 나타난 248명의 동명이인 개체명에는 총 33개 이름의 동명이인이 나타났고, 33개의 이름은 수집된 트윗 데이터에서는 평균 3.45명의 동명이인이 관측되었고 위키피디아 문서 내에서는 평균적으로 4.75명의 동명이인이 관측되었다. 이슈모델에서 사용되는 뉴스기사는 중의성 개체명을 포함하는 트윗이 등록된 날짜를 기준으로 전 후 3일 동안의 분량을 수집하였고(k=3), α, β 그리고 γ값은 실험 데이터에 의존적인 매개 변수로 실험적으로 결정하였다. ($\alpha = 0.4, \beta = 0.35, \Upsilon = 0.25$)

전체 시스템을 위한 지식 베이스는 한국어 위키피디아를 사용하였고, 이슈 모델에서 뉴스기사와 위키피디아 문서의 형태소 분석을 위하여 "Jhannanum"[3] 형태소 분석기를 사용하였다. 또한 실험을 진행하기 전에 모든 트윗 데이터와 위키피디아 문서의 불필요한 데이터들을 삭제하는 전처리 작업을 실시하였다.

표 2에서는 본 논문에서 제안하는 시스템의 성능을 평가하기 위하여 정확도 지표(시스템이 정답으로 판정한 데이터 개수/전체 데이터 개수)를 사용하였으며 각 시스템이 추가 될 때마다 성능을 비교하여 보여준다. 지식 베이스만 사용한 문맥 모델을 베이스라인으로 하여 제안하는 모델들이 각각 추가 되었을 때, 성능이 현저히 증가하였고 최종 시스템 성능은 67.7%을 보였다. 실험 데이터에서 중의성을

2) 본 논문에서는 실험 대상을 동명이인 개체명이 1개만 포함된 트윗들로 제한하였다.

3) Semantic Web Research Center, JHannanum, http://semanticweb.kaist.ac.kr/home/index.php/HanNanum

가지는 개체명이 평균적으로 4.75명의 동명이인을 나타내는 점을 고려하면 높은 정확도를 보였다.

표 2. 시스템 성능 비교

분류	정확도
Contextual model(베이스라인)	0.31
Contextual model + User model	0.59
Context model +User model +Issue model	0.68

표 3에서는 이슈 모델의 뉴스 기사를 정답 위키피디아 페이지에 정확하게 링크하는지 측정한 결과는 보여준다. 수집된 836개 뉴스기사는 2명의 실험자가 교차검증을 통하여 직접 뉴스기사 제목에 나타난 중의성을 가지는 개체명을 태깅하여 실험하고 정확도를 측정한 결과 70.2%의 정확도를 보였다.

표 3. 이슈 모델에서 뉴스 링킹 모듈의 성능

#뉴스기사	#개체명 종류	#나타난 개체명의 수집된 뉴스기사에서의 평균 동명이인	#나타난 개체명의 위키피디아에서의 평균 동명이인	정확도
836개	20명	1.4명	3.35명	0.72

표 4에서 첫 번째 예제에서는 유저 모델을 추가함으로써 중의성 해소의 정확도를 높였음을 알 수 있다. 트윗 사용자가 평소에 야구를 좋아했음을 이전 트윗에서 추출한 “삼성”, “롯데”, “LG”, “야구장” 등의 개체명을 통해 알 수 있고, 그 중 “삼성”과 “롯데” 개체명은 위키피디아 “김태균 (1971)” 페이지에 링크로 나타나므로 중의성 해소에 도움이 되었다. 이슈 모델에서 추가로 추출 된 개체명들이 더욱

표 4. 모델 별 결과 예시

트윗	문맥 모델에서 추출되는 개체명	유저 모델에서 추가 추출되는 개체명	이슈 모델에서 추가 추출되는 개체명
@DooBoo_2 - 김태균이랑 동선이라닠 ㅋㅋㅋㅋㅋ	김태균	삼성, 롯데, LG, 야구장, 조성환,	한국 프로 야구, SK 와이번즈, 삼성 라이온즈, 내야수, 우승 ...
@myhomenamsan - 어제 조인성을 봤다. 보고 난 뒤 화장실에서 거울 봤는데, 우리가 엄마가 잘못했다.	조인성	축구, 일본, 중국...	드라마, 영화, SBS, 배우, 태국, 방콕 ...

“김태균 (1971)”으로 개체 링킹 될 수 있도록 최종 스코어를 높여주었다. 두 번째 예제에서는 첫 번째 예제와 다르게 유저 모델에서는 단서가 되는 개체명을 추출하지 못 하였다. 하지만 이슈 모델이 “드라마”, “영화”, “배우”라는 개체명을 추출함으로써 “조인성”이 “조인성 (배우)”로 링킹 될 수 있도록 하였다.

결 론

본 논문에서는 3가지 스코어링 모델인 문맥 모델, 유저 모델 그리고 이슈 모델을 통합하는 링킹 모델로 이루어진 개체 링킹 시스템을 제안하였다. Charton, E. et al. (2014)의 연구에서 제안한 “mutual relation score” 방법을 한국어 트윗 데이터에서 비교 모델로 채택하였고, 트윗의 특성을 반영하는 유저 모델과 이슈 모델이 추가됨에 따라 성능이 향상되었음을 알 수 있다.

제안한 시스템은 위키피디아와 같은 지식베이스에서 다루지 않는 사소한 사건이나 실시간 사건들을 뉴스 기사를 외부 자원으로 사용하여 한계를 극복하고, 사용자 기록을 사용하여 사용자 관심 분야를 고려했다. 본 논문은 중의성 해소 단계에서 기존의 개체 링킹 방법보다 좋은 성능을 냈지만, 개체명 인식 단계에서 사용

하는 방법은 전통적으로 사용되는 n-gram 용어들 중 개체명 사전에 해당하는 것들만 추출하는 방식을 사용하여 발생하는 오류가 중의성 해소 단계에 서도 적용되어 정확도에 영향을 미치는 현상을 가지고 있다.

향후 과제로는 유저 모델에서 트윗의 해시태그와 같은 자질의 추가에 대해 고려할 것이다. 또한 현재 연구에서는 개체명 인식 시에 좌최장일치법을 사용하고 어절 uni-gram과 bi-gram을 자질로 사용하여 인식하는데, 확률 모델의 사용 등의 추가적인 실험을 통해 성능 향상을 기대 할 수 있다. 이슈 모델에서는 뉴스 기사에서 개체를 뽑는 과정에서의 좀 더 일반적인 규칙과 방법이 필요하며, 뉴스 기사와 위키피디아 문서를 연결하는 방법에 대한 연구가 필요하다. 또한 한국어 이외에 영어 데이터에서도 제안한 방법을 적용할 계획이다.

참고문헌

김영식, 함영균, 김지성, 황도삼, 최기선 (2014). 한국어 텍스트의 개체 URI 탐지: 품사 태깅 독립적 개체명 인식과 중의성 해소, **제26회 한글 및 한국어 정보처리 학술대회 논문집,** 100-106.

이동혁 (2008). 신문기사 속 작은따옴표의 기능. **우리말연구, (23),** 139-162.

Bansal, R., Panem, S., Gupta, M. & Varma, V. (2014). EDIUM: Improving Entity Disambiguation via User Modeling. *Journal of Advances in Information Retrieval, 8416,* 418-423.

Bunescu, R. C. & Pasca, M. (2006). Using Encyclopedic Knowledge for Named entity Disambiguation. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, 6, 9-16.

Charton, E., Meurs, M. J., Jean-Louis, L. & Gagnon, M. (2014). Mutual Disambiguation for Entity Linking. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics,* 476-481.

Cucerzan, S. (2007). 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, 7, 708-716.

Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R.,Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. *Journal of Information Processing and Management 51,* 32-49

Java, A., Song, X., Finin, T. & Tseng, B. (2007). Why We Twitter: Understanding Microblogging Usage and Communities. *Proceedings of the 9th WebKDD and 1st SNA-KDD workshop on Web mining and social network analysis,* 56-65.

Shen, W., Wang, J., Luo, P. & Wang, M. (2013). Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling. *Proceedings of the 19th SIGKDD international conference on Knowledge Discovery and Data mining*, 68-76.

Kang, S., Kim, H., Kang, H. K. & Seo, J. (2014). Lightweight morphological analysis model for smart home applications based on natural language interfaces. *International Journal of Distributed Sensor Networks, 2014,* 1-9.

Liu, X., Zhang, S., Wei, F. & Zhou, M. (2011). Recognizing Named Entities in Tweets. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* 1, 359-369.

Han, X. & Zhao, J. (2009). Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge. *Proceedings of the 18th Conference on Information and Knowledge Management,* 215-224.

Mihalcea, R. & Csomai, A. (2007). Wikify!: Linking Documents to Encyclopedic Knowledge. *Proceedings of the 16th conference on Conference on Information and Knowledge Management,* 233-242.

Milne, D. & Witten, I. H. (2008). Learning to Link with Wikipedia. *Proceedings of the 18th Conference on Information and Knowledge Management,* 215-224.

1차원고접수 : 2015. 10. 01
1차심사완료 : 2015. 11. 17
2차원고접수 : 2015. 11. 27
2차심사완료 : 2015. 12. 03
최종게재승인 : 2015. 12. 03

(*Abstract*)

Entity Linking For Tweets Using User Model and Real-time News Stream

Soyoon Jeong Youngmin Park Sangwoo Kang Jungyun Seo

Computer Science and Engineering Sogang University

Recent researches on Entity Linking(EL) have attempted to disambiguate entities by using a knowledge base to handle the semantic relatedness and up-to-date information. However, EL for tweets using a knowledge base is still unsatisfactory, mainly because the tweet data are mostly composed of short and noisy contexts and real-time issues. The EL system the present work builds up links ambiguous entities to the corresponding entries in a given knowledge base via exploring the news articles and the user history. Using news articles, the system can overcome the problem of Wikipedia coverage (i.e., not handling real-time issues). In addition, given that users usually post tweets related to their particular interests, the current system referring to the user history robustly and effectively works with a small size of tweet data. In this paper, we propose an approach to building an EL system that links ambiguous entities to the corresponding entries in a given knowledge base through the news articles and the user history. We created a dataset of Korean tweets including ambiguous entities randomly selected from the extracted tweets over a seven-day period and evaluated the system using this dataset. We use accuracy index(number of correct answer given by system/number of data set) The experimental results show that our system achieves a accuracy of 67.7% and outperforms the EL methods that exclusively use a knowledge base.

Key words : Named Entity Linking, Entity Linking, Entity Disambiguation, Twotter, Wikipedia