KIPS Trans. Softw. and Data Eng.
Vol.8, No.3 pp.129~136 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2019.8.3.129

A Global-Interdependence Pairwise Approach to Entity Linking Using RDF Knowledge Graph

Yongsun Shim† · Sungkwon Yang†† · Hong-Gee Kim†††

ABSTRACT

There are a variety of entities in natural language such as people, organizations, places, and products. These entities can have many various meanings. The ambiguity of entity is a very challenging task in the field of natural language processing. Entity Linking(EL) is the task of linking the entity in the text to the appropriate entity in the knowledge base. Pairwise based approach, which is a representative method for solving the EL, is a method of solving the EL by using the association between two entities in a sentence. This method considers only the interdependence between entities appearing in the same sentence, and thus has a limitation of global interdependence. In this paper, we developed an Entity2vec model that uses Word2vec based on knowledge base of RDF type in order to solve the EL. And we applied the algorithms using the generated model and ranked each entity. In this paper, to overcome the limitations of a pairwise approach, we devised a pairwise approach based on comprehensive interdependency and compared it.

Keywords : Entity Linking, RDF Knowledgebase, Global-Interdependence

개체 링킹을 위한 RDF 지식그래프 기반의 포괄적 상호의존성 짝 연결 접근법

심 용 선† · 양 성 권†† · 김 홍 기†††

요 약

자연어 표현에는 인물, 조직, 장소, 제품 등의 다양한 개체들이 존재한다. 이러한 개체는 다양한 의미를 가질 수 있다. 이러한 개체가 갖는 중의성 문제는 자연어 처리 분야에 있어 매우 도전적인 과제이다. 개체 링킹(Entity Linking)이란 텍스트에 등장한 개체명을 지식베이스 내의 적절한 개체로 연결해주는 작업이다. 개체 링킹을 위한 대표적인 방법론인 짝 연결 접근법(Pairwise based method)은 한 문장에서 등장한 개체가 두 개 이상일 경우 서로의 연관성을 이용해 개체 링킹을 하는 방법이다. 이 방법은 동일 문장에서 등장하는 개체들 간의 상호의존성(interdependence)만을 고려하고 있어 포괄적인 상호의존성(Global interdependence)이 부족하다는 한계를 갖고 있다. 본 논문에서는 개체 링킹을 위해 RDF 형태의 지식베이스 정보를 바탕으로 Word2vec을 활용한 Entity2vec 모델을 생성하였다. 그리고 생성된 모델을 사용하여 각 개체에 대한 랭킹을 하였다. 본 논문에서는 짝 연결 접근법의 한계점을 보완하기 위해 포괄적인 상호의존성을 바탕으로 짝 연결 접근법을 고안하고 구현 및 실험을 통해 기존의 짝 연결 접근법과 비교하였다.

키워드 : 개체 링킹, RDF 지식베이스, 포괄적 상호의존성

1. 서 론

자연어 표현에는 인물, 조직, 장소, 제품 등의 다양한 개체들이 존재한다. 이러한 개체는 다양한 의미를 가질 수 있다. 이러한 개체가 갖는 중의성 문제는 자연어 처리 분야에 있어 매우 도전적인 과제이다.

개체 링킹(Entity Linking)이란 텍스트에 등장한 개체를 지식베이스 내의 적절한 개체로 연결해주는 작업이다[1]. 이러한 개체 링킹은 질의응답 시스템, 정보추출 시스템 등의 분야에 활용되고 있다. 예를 들어 '영화 레옹에서 마틸다 역할은 내털리 포트먼인가요?'라는 질문에서 사용된 '레옹'은 영화 '레옹'을 의미한다. 이에 반해 '2015년에 발매된 노래인 레옹은 아이유와 박명수가 불렀어?'라는 질문에서 사용된 '레옹'은 2015년 발

※ 본 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2013-0-00109, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발).
† 준 회 원 : 서울대학교 치의과학과 석·박사통합과정
†† 비 회 원 : 서울대학교 의료정보학과 박사과정
††† 정 회 원 : 서울대학교 치의과학과 교수
Manuscript Received : November 29, 2018
First Revision : January 2, 2019
Accepted : January 4, 2019
* Corresponding Author : Hong-Gee Kim(hgkim@snu.ac.kr)

매된 노래 '레옹'을 의미한다. 이와 같이 두 가지 이상의 의미를 가진 개체는 문장에서 동시에 사용된 단어들의 의미와 연관이 있는 것으로 해당 개체의 의미가 결정 된다.

본 논문에서는 개체 링킹을 위해 RDF 형태의 지식베이스를 바탕으로 모델을 생성하였다. 그리고 생성된 모델을 사용하여 개체 링킹을 위한 랭킹을 시도하고, 결과를 비교하였다.

본 논문에서는 개체 링킹을 위한 모델로 Word2vec을 활용한 Entity2vec 모델을 생성하였다. Word2vec은 2013년 구글에서 발표한 연구로, 단어를 벡터화 시키는 워드 임베딩(Word Enbedding)의 방법론 가운데 하나이다[2]. 본 논문에서는 Entity2vec 모델을 생성하기 위해 Word2vec 알고리즘의 모델 학습 기법을 이용하였다. Entity2vec 모델은 지식베이스에 있는 각 개체 간의 연관성을 학습하여 각 개체에 대한 임베딩 벡터로 구성 되어 있다. 그러므로 Entity2vec의 학습데이터는 Word2vec과 달리 문장이 아닌 RDF 지식베이스를 사용하였다. 자연어 문장에는 해당 개체의 의미를 파악하기 위해 필요한 단어들도 있지만, 자연어이기에 불필요한 단어 또한 포함 될 수 있다. 그러나 RDF 지식베이스는 해당 개체의 의미를 파악하기 위해 다른 개체등과의 관계성을 바탕으로 구성되어 있어 자연어 문장이 갖는 이러한 단점을 보완할 수 있다. 그러므로 본 논문에서는 모델 생성에 있어 문장이 아닌 RDF 지식베이스를 학습시켜 각 개체명에 대한 임베딩 정보가 있는 Entity2vec 모델을 생성하였다.

개체 링킹을 위한 대표적인 방법론인 짝 연결 접근법(Pairwise based method)은 한 문장에서 등장한 개체가 두 개 이상일 경우 서로의 연관성을 이용해 개체 링킹을 하는 방법이다[1]. 즉, 두 개의 서로 다른 개체를 연결시킨 뒤 가중치를 부여하고 가중치가 가장 높은 짝을 선택하는 방법이다. 이 방법은 동일 문장에서 등장하는 개체들 간의 상호의존성(interdependence)을 고려하고 있으나, 2개의 짝만을 연결해서 사용하기 때문에 포괄적인 상호의존성(Global interdependence)을 고려하지 못한다는 한계점이 있다. 예를 들어 '십장생의 종류에 소나무도 포함되나요?'라는 질문이 있을 때, 질문에서의 개체는 '십장생', '종류', '소나무'가 추출된다. 중의성 단어인 '소나무'에 대해 전통적인 짝 연결 접근법은 '십장생' 정보가 아닌 바로 이전 단어 '종류' 정보를 사용하여 대한민국의 7인조 여성 음악 그룹인 '소나무'와 연결이 된다. 그러나 본 예제 질문에서 사용된 '소나무'는 나무 '소나무'를 의미하기 때문에 잘못 연결이 됐다 할 수 있다. 본 논문에서는 이러한 전통적인 짝 연결 접근법의 한계점을 보완하기 위해 포괄적인 상호의존성을 바탕으로 짝 연결 접근법을 고안하고 구현 및 실험을 통해 전통적인 짝 연결 접근법과 비교하였다.

본 논문에서는 포괄적인 상호의존성 바탕의 짝 연결 접근법을 사용하여 전통적인 짝 연결 접근법의 한계점을 극복하고자 하였다. 그리고 포괄적인 상호의존성의 개념을 차용한 개별화된 페이지 랭크 알고리즘(Personalized Pagerank Algorithm)과 본 논문에서 제시하는 알고리즘을 비교하였다. 페이지 랭크 알고리즘은 대상이 되는 페이지와 연결되어 있는 다른 페이지의 상대적 중요도에 따라 가중치를 부여하는 알고리즘이다[3]. 본 논문에서는 페이지랭크 알고리즘의 개념적 의미를 차용하여 각 개체를 하나의 페이지로 가정하고 함께 등장한 개체들과의 연관성을 적용한 개별화된 페이지랭크 알고리즘을 사용하였다.

본 논문에서는 개체 링킹을 위해 포괄적인 상호의존성을 바탕으로 제시하는 짝 연결 접근법이 전통적인 짝 연결 접근법뿐만 아니라 개별화된 페이지 랭크 알고리즘을 적용한 결과에 비해 결과가 더 우수한 것을 확인 할 수 있었다.

본 논문은 질의응답 시스템에 적용하기 위해 고안된 것으로, 평가를 위한 테스트 데이터로는 '예/아니오' 정답 형태에 대한 질문셋을 대상으로 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 개체 링킹의 관련 연구를 기술한다. 3장에서는 본 논문에서 사용한 방법론을 기술하고, 4장에서는 실험 결과 및 평가를 기술하였다. 5장에서는 결론 및 차후 연구 방향을 기술하였다.

2. 관련 연구

개체 링킹에 대한 연구는 지난 10년간 활발히 진행되어 왔다. 문장이 있을 때, 개체로 인신되는 단어의 의미와 입력 문장과의 유사도를 비교해서 유사도가 높은 후보군을 출력하는 방식으로 개체 링킹을 하는 지역 호환성 방법(Local compatibility based method)을 사용한 논문들[4-7]은 공통적으로 문장에서 함께 사용된 다른 개체 후보들과의 상호의존성을 충분히 활용하지 못한다는 한계점이 있다.

전통적인 짝 연결 접근법을 사용한 논문들[1, 8, 9]은 지역 호환성 접근법에 비하여 상호의존성을 고려하여 개체 링킹을 하였다. 그러나 전통적인 짝 연결 접근법은 포괄적인 상호의존성을 고려하지 못하고 둘의 관계성만을 사용하는 한계점이 있다. [10] 논문은 로컬 특성, 사용자 특성, 글로벌 특성의 방법을 사용하는데, 이 중 글로벌 특성에 초점을 맞춰 개체 링킹을 위한 방법을 제시하였다.

Word2vec은 단어를 수치화하여 벡터 공간으로 변환시키는 워드 임베딩 방법론 가운데 하나이다. Word2vec은 단어를 임베딩 시키는 다른 방법론들 가운데 성능이 우수한 것으로 알려져 있어 많은 연구에 활용되고 있다. [11] 논문은 RDF 지식베이스에 무작위 걸음(Random walk) 알고리즘을 사용해 학습데이터를 단순 트리플이 아닌 확장된 트리플 형태로 학습 데이터를 구축하고, 개별화된 페이지 랭크 알고리즘과 사전(Prior) 값을 사용해 개체 링킹을 위한 틀(Framework)을 만들었다. [12] 논문은 한국어 서술어와 지식베이스의 프로퍼티를 연결하기 위해 네 가지 자질 값 Availability, Frequency Score, Jaccard Similarity, Word Embedding Similarity의 Weighted Score를 사용하였다. [13] 논문은 단어와 개체를 학습데이터로 Skip-gram을 사용하여 임베딩 모델을 생성한 뒤, Textual

contest similarity, Coherence를 바탕으로 개체 링킹을 하였다. [14] 논문은 Entity Embedding, Contextual attention mechanism, Adaptive local score combination을 주요 요소로 그래픽 모델 및 확률적인 Mention-Entity 맵과 결합하여 개체 링킹을 하였다.

[15] 논문은 한국어 위키피디아 지식베이스를 기반으로 개체 이름 사전을 구축한 뒤, 문맥 유사도, 의미적 관련성, 단서 단어 점수, 개체 표현의 개체명 타입 유사도, 개체 이름 매칭 점수, 개체 인기도 점수를 자질 값으로 서포트 벡터 머신(Support Vector Machine)을 학습하여 NIL 개체를 인식하는 문제와 개체 링킹을 하였다. [16] 논문은 한국어 위키피디아 지식베이스를 이용해 개체 간의 의미관련도를 기반으로 하여 개별적/집단적 개체중의성해소 기법을 사용한 개체 링킹을 하였다.

본 논문에서는 짝 연결 접근법의 개념을 차용하여 알고리즘을 구현하고, 본 논문에서 제시하는 포괄적인 상호의존성을 고려한 짝 연결 접근법과의 비교를 통해 포괄적인 짝 연결 접근법과 전통적인 짝 연결 접근법 그리고 개별화된 페이지 랭크 알고리즘의 성능 차이를 파악하였다.

3. Entity2vec을 활용한 개체 링킹

3장에서는 본 논문의 전체적인 과정에 대해 기술하였다. 과정은 총 3단계로 나뉘어져 있고, Fig. 1에 전체적인 과정이 표현되어 있다. 1단계에서는 Word2vec 알고리즘을 활용한 Entity2vec 모델을 생성하였다. 2단계에서는 문장에서 사용된 단어를 지식베이스에서 검색하여 해당 개체에 대한 후보 개체군을 추출하였다. 마지막 3단계는 개체 링킹 단계로 2단계에서 추출한 전체 후보군에서 각각의 후보를 조합한다. 구축된 조합을 대상으로 Entity2vec 모델에 있는 임베딩 된 벡터를 사용하여 각 후보 간의 유사도를 기반으로 개체 링킹을 하였다. 각 단계에 대한 자세한 내용은 아래에 기술되어 있다.

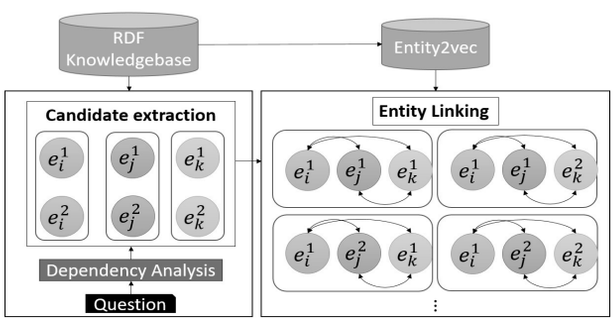


Fig. 1. Overview of Entity Linking Using Entity2vec Model

3.1 Entity2vec 모델 생성

RDF는 Resource Description Framework의 약자로 월드 와이드 웹 컨소시움(World Wide Web Consortium)에서 메타데이터를 모델링하기 위해 제안하였다[17]. RDF 지식베이스는 주어(Subject)-서술어(Property)-목적어(Object)인 트리플(triple) 형태로 되어있다. RDF 지식베이스는 개체 간의 관계를 서술어로 표한하고 있어, 직관적으로 개체 간의 관계를 파악하는데 용이하다. 그러므로 이러한 RDF 지식베이스의 트리플 형태는 자연어 문장으로 풀어 쓸 수 있다. 예를 들어, '레옹-개봉년도-1994년'이라는 트리플이 있을 때, 이 트리플은 '레옹의 개봉년도는 1994년이다.'와 가튼 자연어 문장으로 풀어 쓸 수 있다.

본 논문에서는 RDF 지식베이스를 바탕으로 Word2vec 알고리즘을 사용해 Entity2vec 모델을 만들었다. 단어를 벡터화 시키는 워드 임베딩의 방법론 가운데 하나인 Word2vec은 한 문장에서 해당 단어와 동시에 등장하는 단어들을 학습하여 각 단어에 대한 벡터를 추출한다[2]. Entity2vec 모델은 Word2vec의 개념을 인용하여 자연어 문장으로 표현이 가능한 RDF 지식베이스를 학습데이터로 사용하였다. 즉, 개체들 간의 동시정보(Co-occurence)를 학습하여 각 개체에 대한 벡터를 생성하는 것이다. 이때, 지식베이스에 있는 다양한 트리플 형태 가운데 개체와 개체 간의 관계를 표현하고 있는 형태의 데이터만을 추출하여 학습데이터로 사용하였다. 다시 말하면, 주어와 목적어의 데이터 타입이 URI 형태인 데이터만을 추출해 학습을 시켰다. 이는 URI 형태가 아닌 Datatype의 형태로 되어있는 데이터는 본 논문에서 정의하는 개체라 할 수 없어 데이터 학습에 방해가 될 수 있기에 제외하였다.

3.2 후보군 추출

2단계에서는 문장에서 사용된 단어들 가운데 개체로 인식할 수 있는 단어들을 대상으로 지식베이스에 연결시켜 각 단어의 개체 후보군을 선정했다. 이 경우 개체로 인식 할 수 있는 단어란 문장에서 등장하는 전체 단어 가운데 품사가 명사, 동사인 단어를 의미한다. 예를 들어 '영화 레옹에서 마틸다 역할은 내털리 포트먼인가요?'라는 문장이 있을 때, 해당 문장에 대해 본 논문에서 정의하는 개체로 인식할 수 있는 단어는 '영화', '레옹', '마틸다', '역할', '내털리 포트먼'이다. 해당 단어들은 1차적으로 구문분석(Dependency analysis)을 통해 단어가 가질 수 있는 개체타입을 미리 부여받는다. '레옹, 마틸다, 내털리 포트먼'은 인스턴스, '영화'는 클래스 또는 인스턴스 그리고 '역할'은 프로퍼티로 각각 매핑된다.

본 논문에서는 개체 후보군을 선정하는데 있어 두 가지 정보를 사용하는데, 이때 두 정보 모두 부합했을 경우 개체 후보군으로 선정하였다. 지식베이스에서 단어를 1:1 매칭하여 해당 단어와 매칭 되면서, 동시에 단어의 개체타입과 해당 개체의 개체타입이 매칭 되는 개체를 후보군으로 선정하였다.

예를 들어, 단어 '레옹'의 경우 지식베이스에서 '레옹'으로 검색을 하고, 동시에 인스턴스인 개체만을 후보군으로 선정하였다. 이 경우 후보군은 영화 '레옹', 영화 레옹의 주인공인 캐릭터 '레옹', 노래 제목 '레옹'이 후보군으로 선정되었다.

3.3 개체 링킹

3단계에서는 2단계에서 만들어진 개체 후보군을 대상으로 랭킹을 시행하였다. 랭킹을 위해 각 후보군으로부터 후보들을 추출해 조합(Combination)을 구축하였다. 예를 들어, '영화'의 후보 2개가 추출되고, '레옹'의 후보 3개, '마틸다'의 후보 2개, '역할'의 후보 2개 그리고 '내털리 포트먼'의 후보 1개가 추출되었다고 가정한다. 이때, 조합이란 {영화$_1$, 레옹$_1$, 마틸다$_1$, 역할$_1$, 내털리포트먼$_1$}, {영화$_2$, 레옹$_1$, 마틸다$_1$, 역할$_1$, 내털리포트먼$_1$}과 같이 서로 다른 후보 개체 간의 집합을 의미한다. 위 예제의 경우 총 24개의 조합이 등장한다. 구축된 조합을 대상으로 각 후보군에서 추출된 후보 간의 코사인 유사도를 계산한 뒤 평균을 구한다. 본 논문에서는 전통적인 짝 연결 접근법과 달리 조합 전체의 유사도를 사용함으로써 포괄적인 상호의존성을 바탕으로 개체 중의성을 해소하여 하였다. 포괄적인 상호의존성을 바탕으로 제시하는 짝 연결 접근법의 알고리즘 수식은 (1)과 같다.

$$GIPW = \frac{1}{|V|}\sum_{u,v\in(V\setminus V_k)}\cos(\overrightarrow{(e_u^i)},\overrightarrow{(e_v^j)}) \quad (1)$$

Equation (1)은 조합의 스코어를 의미한다. Equation (1)은 단어의 개수에 따라 V개의 노드셋(Node set)을 갖는다. 노드셋은 한 단어에 대한 후보 개체의 집합을 의미한다. e_u^i는 u 노드셋의 i 번째 노드 개체를 의미하고, e_v^j는 v 노드셋의 j 번째 노드 개체를 의미한다. 즉, Equation (1)은 조합에 있는 개체 후보 간 짝 지을 수 있는 모든 경우의 수를 대상으로 코사인 유사도를 계산하여 모두 합한 뒤, 단어의 개수로 나누어 조합의 평균값을 구한다. 이후 전체 조합을 대상으로 값을 구한 뒤, 그 중 가장 큰 값을 가진 조합을 선택하여 개체 링킹을 하였다.

Fig. 2는 {영화$_1$,레옹$_1$,마틸다$_1$,역할$_1$,내털리포트먼$_1$} 조합이 있을 때 Equation (1)을 적용시킨 예시이다.

$$GIPW_{\text{영화}_1,\text{레옹}_1,\text{마틸다}_1,\text{역할}_1,\text{내털리포트먼}_1}$$
$$= \frac{1}{5}\{\cos(\overrightarrow{\text{영화}_1},\overrightarrow{\text{레옹}_1}) + \cos(\overrightarrow{\text{영화}_1},\overrightarrow{\text{마틸다}_1}) + \cdots + \cos(\overrightarrow{\text{역할}_1},\overrightarrow{\text{내털리포트먼}_1})\}$$

Fig. 2. Example of Applying GIPW Algorithm

4. 실험 및 평가

본 논문에서는 Entity2vec 모델 생성을 위한 학습 데이터로 Adam 지식베이스를 사용하였다. Adam 지식베이스는 지식베이스 기반 질의응답 시스템 개발을 위해 구축한 RDF 지식베이스로, 약 1700만개의 인스턴스, 1천개의 프로퍼티, 2억개의 트리플로 구성되어 있다[18]. Adam 지식베이스는 Table 1과 같이 구성되어 있다. 각 괄호 안에 있는 번호는 해당 인스턴스 혹은 프로퍼티의 고유 ID를 의미한다.

Table 1. Examples of Adam Knowledgebase

Subject	Property	Object
레옹(0000093989)	감독(director)	뤼크 베송(0000133848)
레옹(0000093989)	생성일(startedOn)	1995-02-18(1995-02-18)
레옹(0030550071)	연기자(player)	장 르노(0000545064)

본 논문은 질의응답 시스템에 적용하기 위해 고안되었으므로, 평가를 위한 테스트 데이터로는 솔트룩스에서 제공한 질문 셋을 사용하였다. 질문 셋은 '아이린은 걸그룹 레드벨벳의 멤버야?', '영화 레옹에서 마틸다 역할은 내털리 포트먼인가요?'와 같이 정답이 '예/아니오'의 형태인 질문들로 구성되어 있다. 본 논문에서는 총 198개의 질문 가운데 구문분석이 정확하게 이뤄지지 않아 단어 추출에 오류가 있는 질문 51개 그리고 지식베이스 내의 단어가 부족해 후보군을 찾을 수 없는 질문 53개를 제외한 94개의 질문을 대상으로 실험을 진행하였다. 94개의 질문에 대해 구문분석을 하면 총 370개의 단어와 단어의 개체 타입 정보가 추출되었다. 370개의 단어 당 지식베이스 내에 있는 중의성 개체의 후보는 평균 4.2개였다.

시스템을 평가하기 위해 370개의 단어에 대해 수동으로 정답 셋을 구축하였다. 정답 셋의 형태는 Table 2와 같다.

Table 2. Examples of Answer Set

Word	Entity type	Answer
내털리 포트먼	Resource	http://kb.adams.ai/resource/0000410364
레옹	Resource	http://kb.adams.ai/resource/0000093989
영화	Class	http://kb.adams.ai/schema/class/movie_06205452
역할	Property	http://kb.adams.ai/schema/property/role
마틸다	Resource	http://kb.adams.ai/resource/0030163951

본 논문에서는 상호의존성만을 고려한 짝 연결 접근법(Interdependence based Pairwise, IPW), 포괄적인 상호의존성을 고려한 짝 연결 접근법(Global Interdependence based Pairwise, GIPW), 포괄적인 상호의존성을 바탕으로 고안된 개별화된 페이지랭크 알고리즘(Personalized Pagerank, PPR)을 비교하여 테스트를 진행하였다.

IPW 알고리즘은 개체 간의 상호의존성만을 고려하기 때문에, 조합에 있는 개체 후보의 순서에 따라 코사인 유사도를 계산하여 모두 합한 뒤, 단어의 개수로 나누어 평균값을 구한

다. 이후 전체 조합을 대상으로 값을 구한 뒤, 가장 큰 값을 가진 조합을 선택하여 개체 링킹을 하였다. IPW 알고리즘 수식은 (2)와 같다.

$$IPW = \frac{1}{|V|}\sum_{u(=V_k),v(=V_{k+1})\in(V\setminus V_i)}\cos(\overrightarrow{(e_u^i)},\overrightarrow{(e_v^j)}) \quad (2)$$

Fig. 3은 {영화$_1$,레옹$_1$,마틸다$_1$,역할$_1$,내털리포트먼$_1$} 조합이 있을 때 Equation (2)을 적용시킨 예시이다. GIPW와 달리 조합 내에 있는 개체 후보가 등장한 순서대로 짝을 지어 코사인 유사도를 계산한다.

PPR의 경우 한 노드에서 다른 노드로 이동 할 때 엣지에 가중치를 부여하기 위한 알고리즘으로는 [11]의 Entity Transition Probabilities (ETP)알고리즘을 사용하였다. 이 알고리즘은 단어의 개수에 따라 V 개의 노드셋을 갖는데, 노드셋은 한 단어에 대한 후보 개체의 집합을 의미하고, 노드는 노드셋 안에 있는 k 개의 후보를 의미한다. 알고리즘 수식은 (3)과 같다.

$$\begin{aligned}&IPW_{\text{영화}_1,\text{레옹}_1,\text{마틸다}_1,\text{역할}_1,\text{내털리포트먼}_1}\\&=\frac{1}{5}\{\cos\left(\overrightarrow{\text{영화}_1},\overrightarrow{\text{레옹}_1}\right)+\cos\left(\overrightarrow{\text{레옹}_1},\overrightarrow{\text{마틸다}_1}\right)+\cos\left(\overrightarrow{\text{마틸다}_1},\overrightarrow{\text{역할}_1}\right)\\&+\cos\left(\overrightarrow{\text{역할}_1},\overrightarrow{\text{내털리포트먼}_1}\right)\}\end{aligned}$$

Fig. 3. Example of Applying IPW Algorithm

$$ETP(e_u^i e_v^j) = \frac{\cos(\overrightarrow{(e_u^i)},\overrightarrow{(e_v^j)})}{\sum_{k\in(V\setminus V_i)}\cos(\overrightarrow{(e_u^i)},\overrightarrow{(k)})} \quad (3)$$

ETP알고리즘의 적용 예시는 Fig. 4를 참고하면 된다. Fig. 4는 문장에 사용된 '레옹'과 '내털리포트먼'을 대상으로 ETP 알고리즘을 적용시켰다. '레옹'이라는 단어로 만들어진 노드셋이 있고, 그 밑에 노드로 영화 제목 '레옹', 캐릭터 '레옹', 노래제목 '레옹' 총 3개의 개체후보가 있고, '내털리포트먼'이라는 단어로 만들어진 노드셋이 있고, 그 밑에 노드로 배우 '내털리포트먼' 총 1개의 개체후보가 있다. Entity2vec 모델에 있는 각 노드의 벡터를 ETP 알고리즘에 적용시키면, 배우 '내털리포트먼'은 '레옹' 노드셋에 있는 노드들로부터 각각 1점을 받게 된다. 반대로, 배우 '내털리포트먼' 노드셋은 다른 노드에 부여할 수 있는 총 1점을 '레옹' 노드셋에 있는 노드들에 각각 나눠 점수를 부여하게 되는데, 마찬가지로 ETP 알고리즘에 적용시키면, 영화 '레옹'은 0.4점, 캐릭터 '레옹'은 0.4점, 노래제목 '레옹'은 0.2점을 받는다. 이와 같은 방식으로 두 개의 노드셋이 서로의 노드 간의 ETP 알고리즘을 적용한 값을 누적하여 최종적으로 각각의 노드셋에서 가장 점수가 높은 노드를 정답으로 개체 링킹을 하였다.

본 논문에서는 구문 분석 및 지식베이스 내 단어의 부족으로 인해 후보군을 찾을 수 없는 단어가 있는 질문을 제외했기 때문에, 본 논문에서 평가하는 데이터는 지식베이스 내 후보군이 존재하는 단어가 대상이 되므로 기존의 시스템들과 달리 정밀도(Precision), 재현율(Recall)이 아닌 정확률(Accuracy)로 평가하였다.

세 가지 방법론의 테스트 결과는 Table 3에 있다. Table 3는 Entity2vec 모델을 학습하는데 있어, DL4J에서 제공하는 Word2vec 알고리즘의 기본 옵션(Baseline)을 사용한 결과이다 [19]. 정확률은 IPW가 63.2%, GIPW가 65.7%, PPR이 64.3%로 포괄적인 상호의존성을 고려한 GIPW와 PPR이 IPW보다 결과가 높았고, 그 중 GIPW가 PPR보다 1.4% 높았다. 이는 상호의존성만을 고려한 IPW에 비해 포괄적인 상호의존성을 고려한 방법론이 개체 링킹에 있어 더 효과적인 것을 알 수 있었고, 그 중 본 논문에서 제안한 포괄적인 상호의존성을 고려한 짝 연결 접근법이 개체 링킹에 있어 효과적인 것을 확인할 수 있었다.

Table 3. Test Results of IPW, GIPW, PPR

	Baseline		
	IPW	GIPW	PPR
Accuracy	63.2%	**65.7%**	64.3%

Table 4는 기본 옵션에서 Iteration을 1에서 10으로 늘려 Entity2vec 모델을 생성 후 테스트한 결과이다. Iteration을 1에서 10으로 늘려 학습한 모델을 적용한 결과 IPW가 정확률 67%, GIPW가 70.5%, PPR이 65.6%로 GIPW가 IPW와 PPR에 비해 효과적인 것을 확인 할 수 있었다.

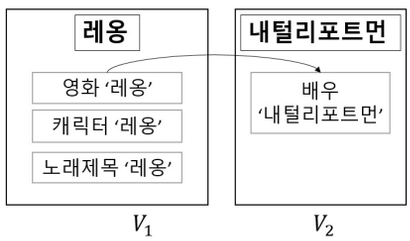
Table 4. Test Results of a Change of Iteration

	Iteration increase (1 -> 10)		
	IPW	GIPW	PPR
Accuracy	67%	**70.5%**	65.6%

Table 5는 기본 옵션에서 Epoch을 1에서 10으로 늘려 Entity2vec 모델을 생성 후 테스트한 결과이다. Epoch을 1에서 10으로 늘려 학습한 모델을 적용한 결과 GIPW와 PPR이 정확률 67.3%로 동일한 결과가 나온 것을 볼 수 있었다.

Table 5. Test Results of a Change of Epoch

	Epoch increase (1 -> 10)		
	IPW	GIPW	PPR
Accuracy	66.5%	**67.3%**	**67.3%**



$$ETP(\text{영화'레옹'}, \text{배우'내털리포트먼'}) = \frac{\cos(\text{영화·레옹}, \text{배우·내털리포트먼})}{\cos(\text{영화·레옹}, \text{배우·내털리포트먼})+\cos(\text{캐릭터·레옹}, \text{배우·내털리포트먼})+\cos(\text{노래제목·레옹}, \text{배우'내털리포트먼})}$$

Fig. 4. Example of Applying ETP Algorithm

Table 6은 기본 옵션에서 Layersize를 100에서 200으로 늘려 Entity2vec 모델을 생성 후 테스트한 결과이다. Layersize를 100에서 200으로 늘려 학습한 모델을 적용한 결과 GIPW의 정확률 65.1%로 IPW와 PPR에 비해 효과적인 것을 볼 수 있다. 그러나 기본 옵션과 비교해보면 결과가 낮아진 것을 확인 할 수 있었다.

Table 6. Test Results of a Change of Layersize

	Layersize increase (100 -> 200)		
	IPW	GIPW	PPR
Accuracy	62.4%	65.1%	64.6%

Iteration, Epoch, Layersize를 각각 늘린 모델의 결과 가운데 Iteration만을 늘려 생성한 모델의 결과가 가장 우수한 것으로 나타났다. 그리고 Iteration과 Epoch을 증가시켜 학습한 모델의 성능이 기본 옵션으로 학습시킨 모델에 비해 효과가 우수한 것을 확인 할 수 있었다. 이는 곧, Iteration과 Epoch 옵션을 증가시켜 모델을 학습시키는 것이 Layersize를 증가시켜 모델을 학습시키는 것에 비해 보다 발전된 모델을 만드는데 있어 중요한 요소인 것을 확인 할 수 있었다.

5. 결 론

본 논문에서는 Word2vec을 사용해서 Entity2vec 모델을 생성하고 임베딩 된 벡터를 이용해 개체명 중의성을 해소하였다. 본 논문에서는 각 단어의 상호의존성을 고려한 접근법보다는 전체 단어를 활용하는 포괄적인 상호의존성을 고려한 접근법이 전체적으로 우수한 것을 볼 수 있었다. 포괄적인 상호의존성을 고려한 접근법 가운데 개별화된 페이지랭크 알고리즘을 적용한 것 보다 짝 연결 접근법이 우수한 것을 확인할 수 있었다.

기존의 개체 링킹 연구에서는 개체를 인스턴스와 프로퍼티 중 하나만을 대상으로 이루어졌지만, 본 연구에서는 인스턴스와 프로퍼티를 모두 개체로 인식하여 확장된 알고리즘을 개발했다는 점에서 확장성이 매우 높다 할 수 있다. 인스턴스와 프로퍼티를 모두 고려한 이전 연구는 영어 데이터를 대상으로 이루어졌다[20]. 본 연구에서는 한글 데이터를 대상으로 인스턴스와 프로퍼티를 모두 고려하여 확장된 연구를 진행했다는 점에서 독창적이라 할 수 있다.

본 논문에서는 Entity2vec 모델을 생성하는데 있어 모델 생성 옵션 변경에 따라 결과가 달리 나타나는 것을 볼 수 있었다. 이는 추후 모델을 생성하는데 있어 Iteration, Epoch 증가의 최적 옵션을 찾는다면 최적화된 모델 생성을 기대할 수 있다.

그리고 본 논문에서의 학습데이터는 주어(Subject)-서술어(Property)-목적어(Object) 형태의 1차원의 트리플 형태였다. 추후 학습데이터를 무작위 걸음(Random walk) 등의 알고리즘을 사용해서 $R_1P_3R_2P_2S_1P_1O_1R_3P_4R_4P_5R_5$과 같이 트리플 차원을 확장시킨다면 해당 개체에 대한 의미를 더 자세히 표현 할 수 있기 때문에 보다 발전된 모델을 생성할 수 있을 것으로 예상한다. 이때, 한 개의 개체에 대해 지식베이스 전체의 정보를 활용하는 것 보다는 무작위 걸음 등의 알고리즘으로 제한을 두고 학습시키면 시간과 비용 소모를 줄이면서 좋은 모델을 만들 수 있을 것으로 예상한다[21].

References

[1] Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabarti, S., "Collective annotation of Wikipedia entities in web text," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining,* ACM, pp.457-466, June, 2009.

[2] Mikolov, T., Chen, K., Corrado, G., and Dean, J. "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781., 2013.

[3] Page, L., Brin, S., Motwani, R., and Winograd, T., "The PageRank citation ranking: Bringing order to the web,"

Stanford InfoLab, 1999.

[4] Bunescu, R. and Pasca, M., "Using encyclopedic knowledge for named entity disambiguation," in *11th conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[5] Cucerzan, S., Large-scale named entity disambiguation based on Wikipedia data. in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[6] Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T., "Entity disambiguation for knowledge base population," in *Proceedings of the 23rd International Conference on Computational Linguistics*, Association for Computational Linguistics, pp.277-285, Aug. 2010.

[7] Fader, A., Soderland, S., Etzioni, O., and Center, T., "Scaling Wikipedia-based named entity disambiguation to arbitrary web text," in *Proceedings of the IJCAI Workshop on User-contributed Knowledge and Artificial Intelligence: An Evolving Synergy*, Pasadena, CA, USA, pp.21-26, Jan. 2009.

[8] Milne, D. and Witten, I. H., "Learning to link with wikipedia," in *Proceedings of the 17th ACM conference on Information and knowledge management*, ACM, pp.509-518, Oct. 2008.

[9] Medelyan, O., Witten, I. H., and Milne, D., "Topic indexing with Wikipedia," in *Proceedings of the AAAI WikiAI Workshop*, Vol.1, pp.19-24, Jul. 2008.

[10] SeoHyun Kim, YoungDuk Seo, and Doo-Kwon Baik, "Tweet Entity Linking Method based on User Similarity for Entity Disambiguation," *Journal of KIISE*, Vol.43, No.9, pp.1043-1051, 2016.

[11] Zwicklbauer, S., Seifert, C., and Granitzer, M, "DoSeR-a knowledge-base-agnostic framework for entity disambiguation using semantic embeddings," in *International Semantic Web Conference*, Springer, Cham, pp.182-198, May 2016.

[12] Wousung Won, Jongseong Woo, Jiseong Kim, YoungGyun Hahm, and Key-Sun Choi, "Linking Korean Predicates to Knowledge Base Properties," *Journal of KIISE*, Vol.42, No.12, pp.1568-1574, 2015.

[13] Yamada, I., Shindo, H., Takeda, H., and Takefuji, Y., "Joint learning of the embedding of words and entities for named entity disambiguation," arXiv preprint arXiv:1601.01343., 2016.

[14] Ganea, O. E. and Hofmann, T. "Deep joint entity disambiguation with local neural attention," arXiv preprint arXiv:1704.04920., 2017.

[15] Hokyung Lee., Jaehyuun An., Jeongmin Yoon., Kyoungman Bae., and Youngjoong Ko., "A Method to Solve the Entity Linking Ambiguity and NIL Entity Recognition for efficient Entity Linking based on Wikipedia," *Journal of KIISE*, Vol.44, No.8, pp.813-821, 2017.

[16] In-Su Kang, "An Effect of Semantic Relatedness on Entity Disambiguation: Using Korean Wikipedia," *Journal of Korean Institute of Intelligent Systems*. Vol.25, No.2, pp.111-118, 2015.

[17] Miller, E. "An introduction to the resource description framework," *Bulletin of the American Society for Information Science and Technology*, Vol.25, No.1, pp.15-19, 1998.

[18] Saltlux's Adam Platform [internet], http://adams.ai/.

[19] Deep Learning for Java [internet], https://deeplearning4j.org/

[20] Dubey, M., Banerjee, D., Chaudhuri, D., and Lehmann, J., "EARL: Joint Entity and Relation Linking for Question Answering over Knowledge Graphs," arXiv preprint arXiv:1801.03825., 2018.

[21] Goyal, P. and Ferrara, E. "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, Vol.151, pp.78-94, 2018.

심 용 선

http://orcid.org/0000-0003-3800-0594

e-mail : yongsun0926@snu.ac.kr

2014년 충북대학교 농업경제학과(학사)

2015년~현 재 서울대학교 치의과학과 석・박사통합과정

관심분야 : Natural Language Processing, Artificial Intelligence, Semantic Web, Question Answering System

양 성 권

http://orcid.org/0000-0001-7253-486x

e-mail : sungkwon.yang@snu.ac.kr

2007년 선문대학교 컴퓨터공학과(학사)

2010년 서울대학교 치의과학과(석사)

2010년~현 재 서울대학교 의료정보학과 박사과정

관심분야 : Knowledge Representation, Semantic Web, Question Answering System

김 홍 기

http://orcid.org/0000-0002-2610-4321

e-mail : hgkim@snu.ac.kr

1985년 고려대학교 심리학과(학사)

1993년 University of Georgia
전산학과(석사)

1996년 University of Georgia
철학과(박사)

2005년~현 재 서울대학교 치의과학과 교수

관심분야 : Artificial Intelligence, Semantic Web, Ontology, Knowledge Representation, Machine Learning