(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0117690
(43) 공개일자 2020년10월14일

(51) 국제특허분류(Int. Cl.)
G06N 5/02 (2006.01) *G06F 16/901* (2019.01)
G06F 17/15 (2006.01)
(52) CPC특허분류
G06N 5/025 (2019.01)
G06F 16/9024 (2019.01)
(21) 출원번호 10-2019-0040173
(22) 출원일자 2019년04월05일
심사청구일자 2019년04월05일

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
이경호
서울특별시 중구 다산로 32, 5동 1002호 (신당동, 남산타운)
오병국
서울특별시 마포구 신촌로12나길 30, 702호 (노고산동)
(74) 대리인
특허법인우인

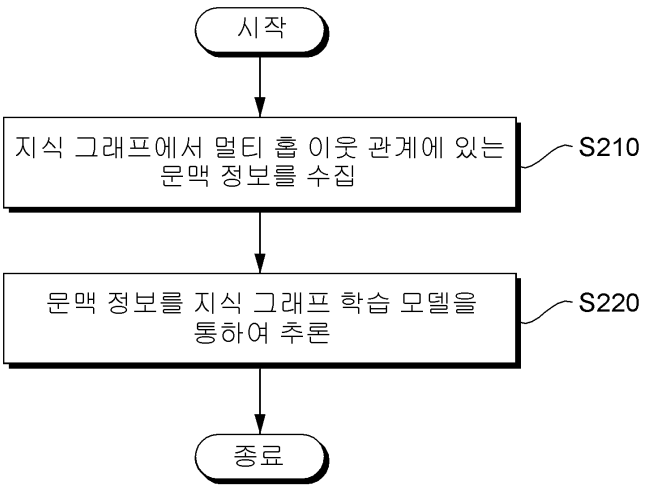
전체 청구항 수 : 총 14 항

(54) 발명의 명칭 **멀티 홉 이웃을 이용한 컨볼루션 학습 기반의 지식 그래프 완성 방법 및 장치**

(57) 요 약

본 실시예들은 다양한 연결 패턴이 포함된 멀티 홉 이웃 관계에 있는 문맥 정보를 컨볼루션 분석함으로써, 누락되거나 부정확한 엔티티 또는 관계를 보다 정확하게 추론할 수 있는 지식 그래프 완성 방법 및 장치를 제공한다.

대 표 도 - 도3



(52) CPC특허분류
G06F 17/15 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 NRF-2016R1A2B4015873
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 한국연구재단 중견연구자지원사업
연구과제명 사물인터넷을 위한 복합 이벤트 처리 기반 개인맞춤 상황인지형 추천 시스템
기 여 율 1/1
과제수행기관명 연세대학교
연구기간 2016.06.01 ~ 2019.05.31

명 세 서

청구범위

청구항 1

컴퓨팅 디바이스에 의한 지식 그래프 완성 방법에 있어서,

지식 그래프에서 멀티 홉 이웃 관계에 있는 문맥 정보를 수집하는 단계; 및

상기 문맥 정보를 지식 그래프 학습 모델을 통하여 추론하는 단계

를 포함하는 지식 그래프 완성 방법.

청구항 2

제1항에 있어서,

상기 문맥 정보는 엔티티를 포함하고 상기 엔티티 간의 관계를 포함하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 3

제1항에 있어서,

상기 문맥 정보를 수집하는 단계는,

상기 지식 그래프에 속하는 루트 엔티티로부터 기 설정된 윈도우 크기 범위 내에서 너비 우선 검색(Breadth First Search)을 수행하여 상기 지식 그래프에 속하는 엔티티를 획득하여 상기 문맥 정보에 해당하는 멀티 홉 이웃 정보에 추가하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 4

제3항에 있어서,

상기 문맥 정보를 수집하는 단계는,

상기 획득한 엔티티의 개수가 상기 윈도우 크기 범위보다 작으면 다음 이웃 엔티티를 상기 멀티 홉 이웃 정보에 추가하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 5

제1항에 있어서,

상기 문맥 정보를 수집하는 단계는,

상기 문맥 정보에 해당하는 멀티 홉 이웃 정보에서 엔티티 간의 거리를 기준으로 고유한 순위를 지정하고 기 설정된 윈도우 크기 범위 내에서 일부의 엔티티를 제거하여 상기 문맥 정보를 정규화하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 6

제1항에 있어서,

상기 문맥 정보를 수집하는 단계는,

상기 문맥 정보에 해당하는 멀티 홉 이웃 정보의 그룹별 순위에 따라 깊이 우선 탐색(Depth First Search)을 수행하여 상기 멀티 홉 이웃 정보를 비선형 공간으로부터 선형 공간으로 매핑하여 상기 문맥 정보를 변환하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 7

제1항에 있어서,

상기 지식 그래프 학습 모델은 엔티티와 관계에 관한 상기 문맥 정보를 복수 차원의 임베딩 행렬에서 문맥 정보 임베딩으로 인코딩하는 제1 레이어를 포함하며,

상기 문맥 정보가 고정된 크기를 갖도록 상기 문맥 정보 임베딩에 가변적 크기를 갖는 패딩 벡터가 추가되는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 8

제7항에 있어서,

상기 지식 그래프 학습 모델은 컨볼루션 필터를 적용하여 상기 문맥 정보로부터 연결 패턴에 관한 문맥 특징 벡터를 추출하는 제2 레이어를 포함하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 9

제8항에 있어서,

상기 지식 그래프 학습 모델은 상기 문맥 특징 벡터와 상기 제1 레이어에 의해 인코딩된 엔티티 임베딩을 벡터 공간에서 결합하는 제3 레이어를 포함하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 10

제9항에 있어서,

상기 지식 그래프 학습 모델은 상기 결합된 문맥 특징 벡터 및 엔티티 임베딩에 상기 제1 레이어에 의해 인코딩된 관계 임베딩을 가중합하여 합성 벡터를 생성하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 11

제10항에 있어서,

상기 지식 그래프 학습 모델은 상기 합성 벡터에 상기 제1 레이어에 의해 인코딩된 후보 엔티티 임베딩을 행렬곱하여 점수를 산출하는 제4 레이어를 포함하는 것을 특징으로 하는 지식 그래프 완성 방법.

청구항 12

하나 이상의 프로세서 및 상기 하나 이상의 프로세서에 의해 실행되는 하나 이상의 프로그램을 저장하는 메모리를 포함하는 지식 그래프 완성 장치에 있어서,

상기 프로세서는 지식 그래프에서 멀티 홉 이웃 관계에 있는 문맥 정보를 수집하고,

상기 프로세서는 상기 문맥 정보를 지식 그래프 학습 모델을 통하여 추론하는 것을 특징으로 하는 지식 그래프 완성 장치.

청구항 13

제12항에 있어서,

상기 프로세서는 상기 지식 그래프에 속하는 루트 엔티티로부터 기 설정된 윈도우 크기 범위 내에서 너비 우선 검색(Breadth First Search)을 수행하여 상기 지식 그래프에 속하는 엔티티를 획득하여 상기 문맥 정보에 해당하는 멀티 홉 이웃 정보에 추가하는 것을 특징으로 하는 지식 그래프 완성 장치.

청구항 14

제12항에 있어서,

상기 지식 그래프 학습 모델은 엔티티와 관계에 관한 상기 문맥 정보를 복수 차원의 임베딩 행렬에서 문맥 정보 임베딩으로 인코딩하는 제1 레이어를 포함하며,

상기 문맥 정보가 고정된 크기를 갖도록 상기 문맥 정보 임베딩에 가변적 크기를 갖는 패딩 벡터가 추가되는 것을 특징으로 하는 지식 그래프 완성 장치.

발명의 설명

기 술 분 야

[0001] 본 실시예가 속하는 기술 분야는 컨볼루션 학습 기반의 지식 그래프 완성 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 이 부분에 기술된 내용은 단순히 본 실시예에 대한 배경 정보를 제공할 뿐 종래기술을 구성하는 것은 아니다.

[0003] 지식 그래프는 효과적인 데이터 모델링 중 하나이다. 지식 그래프는 엔티티 모호성, 질의 응답, 정보 검색 등의 다양한 어플리케이션에서 중요하게 활용되고 있다.

[0004] 지식 그래프는 팩트(fact)이라 불리는 트리플 <h,r,t>의 집합으로 구성되며, 헤드 엔티티 h와 테일 엔티티 t는 의미적 관계 r로 연결되어 있다. 지식 그래프는 개방형 연계 데이터(Linked Open Data)와 같은 거대한 이종의 소스 지식들을 통합하고 상호 운용을 가능하게 한다.

[0005] 예컨대, DBpedia, WikiData, Freebase, WordNet, 및 Google Knowledge Graph와 같은 다양한 지식 그래프들이 생성되었다. 이러한 지식 그래프들은 많은 정보들이 누락되거나 부정확하므로, 아직까지는 실세계에서 적극적으로 활용하는데 한계가 있다.

[0006] 지식 그래프에 관한 최신의 기술들은 지식 그래프 내 엔티티와 관계를 저차원 임베딩 벡터로 변환하기 위해서 크게 두 가지 접근 방법을 활용한다.

[0007] 트리플 <h,r,t>가 주어질 때, 첫 번째 방법인 전이 기반의 모델(translational model)은 관계 벡터가 두 엔티티 벡터 간의 전이 벡터가 되도록 학습을 수행한다. 두 번째 방법인 구성 기반의 모델(compositional model)은 트리플 점수화 함수를 활용하여 관계와 두 엔티티 벡터 구성의 점수를 매기고 존재하는 트리플의 점수가 1 또는 존재하지 않는 트리플의 점수가 0이 되도록 학습한다.

[0008] 최근 상황 정보 활용이 임베딩 벡터의 성능 향상에 중요해짐에 따라 주어진 단일 트리플만을 고려하는 것이 아닌 관련된 다수 트리플을 고려할 필요가 있다. 구체적으로, 주어진 트리플의 두 엔티티를 연결하는 관계 경로 또는 주변 이웃 트리플 정보들을 학습에 고려한다.

[0009] 그러나, 기존의 방법은 주어진 트리플의 주변 연결 패턴들을 제대로 반영하지 못하고, 불필요한 경로 또는 이웃들을 고려하기 때문에 학습을 최적화하는데 한계가 있다.

선행기술문헌

특허문헌

[0010] (특허문헌 0001) 한국등록특허 제10-1914853호 (2018.11.02)

발명의 내용

해결하려는 과제

[0011] 본 발명의 실시예들은 멀티 홉 이웃 관계에 있는 문맥 정보를 분석함으로써, 누락되거나 부정확한 엔티티 또는 관계를 보다 정확하게 추론하는 데 주된 목적이 있다.

[0012] 본 발명의 명시되지 않은 또 다른 목적들은 하기의 상세한 설명 및 그 효과로부터 용이하게 추론할 수 있는 범위 내에서 추가적으로 고려될 수 있다.

과제의 해결 수단

[0013] 본 실시예의 일 측면에 의하면, 컴퓨팅 디바이스에 의한 지식 그래프 완성 방법에 있어서, 지식 그래프에서 멀티 홉 이웃 관계에 있는 문맥 정보를 수집하는 단계, 및 상기 문맥 정보를 지식 그래프 학습 모델을 통하여 추론하는 단계를 포함하는 지식 그래프 완성 방법을 제공한다.

[0014] 본 실시예의 다른 측면에 의하면, 하나 이상의 프로세서 및 상기 하나 이상의 프로세서에 의해 실행되는 하나 이상의 프로그램을 저장하는 메모리를 포함하는 지식 그래프 완성 장치에 있어서, 상기 프로세서는 지식 그래프에서 멀티 홉 이웃 관계에 있는 문맥 정보를 수집하고, 상기 프로세서는 상기 문맥 정보를 지식 그래프 학습 모델을 통하여 추론하는 것을 특징으로 하는 지식 그래프 완성 장치를 제공한다.

발명의 효과

[0015] 이상에서 설명한 바와 같이 본 발명의 실시예들에 의하면, 다양한 연결 패턴이 포함된 멀티 홉 이웃 관계에 있는 문맥 정보를 컨볼루션 모델을 통하여 분석함으로써, 누락되거나 부정확한 엔티티 또는 관계를 보다 정확하게 추론할 수 있는 효과가 있다.

[0016] 여기에서 명시적으로 언급되지 않은 효과라 하더라도, 본 발명의 기술적 특징에 의해 기대되는 이하의 명세서에서 기재된 효과 및 그 잠정적인 효과는 본 발명의 명세서에 기재된 것과 같이 취급된다.

도면의 간단한 설명

[0017] 도 1은 지식 그래프를 예시한 도면이다.

도 2는 본 발명의 일 실시예에 따른 지식 그래프 완성 장치를 예시한 블록도이다.

도 3은 본 발명의 다른 실시예에 따른 지식 그래프 완성 방법을 예시한 흐름도이다.

도 4는 본 발명의 다른 실시예에 따른 지식 그래프 완성 방법에서 문맥 정보를 수집하는 단계를 예시한 흐름도이다.

도 5 내지 도 8은 본 발명의 다른 실시예에 따른 지식 그래프 완성 방법에서 문맥 정보를 수집하는 단계를 예시한 도면이다.

도 9는 본 발명의 다른 실시예에 따른 지식 그래프 완성 방법에서 문맥 정보를 추론하는 단계를 예시한 흐름도이다.

도 10은 본 발명의 실시예들에 따른 지식 그래프 학습 모델을 예시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0018] 이하, 본 발명을 설명함에 있어서 관련된 공지기능에 대하여 이 분야의 기술자에게 자명한 사항으로서 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하고, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다.

[0019] 도 1은 지식 그래프를 예시한 도면이다.

[0020] 도 1을 참조하면, < 대니얼 추이(Daniel Tsui), 에서 교육받다(EducatedAt), 프린스턴 대학(Princeton Univ.) >, < 대니얼 추이(Daniel Tsui), 에서 태어나다(PlaceOfBirth), 뉴저지(New Jersey) > 팩트들이 있다.

[0021] 기존의 지식 그래프 완성 모델은 < 프린스턴 대학(Princeton Univ.), 에 위치하다(LocatedAt), 미국(United States) >, < 뉴저지(New Jersey), 의 주이다(StateOf), 미국(United States) > 팩트들을 고려하지 않으므로, '의 시민이다(CitizenOf) ' 관계를 추론하지 못한다.

[0022] 기존의 지식 그래프 완성 모델은 {입상하다(Awarded), 노벨상(Nobel Prize), 의 수상분야이다(AwardFieldOf)} 과 {에서 교육받다(EducatedAt), 프린스턴 대학(Princeton Univ.), 교육하다(Educate)} 을 고려하지 않으므로, '물리학(Physics)' 엔티티를 추론하지 못한다.

[0023] 본 실시예에 따른 지식 그래프 완성 장치는 멀티 홉 이웃 정보를 고려하고 멀티 홉 이웃 정보와 엔티티를 결합하여 학습함으로써, 점선으로 표시된 '물리학' 엔티티와 점선으로 표시된 '의 시민이다' 관계를 추론할 수 있다.

[0024] 도 2는 본 발명의 일 실시예에 따른 지식 그래프 완성 장치를 예시한 블록도이다.

[0025] 지식 그래프 완성 장치(110)는 적어도 하나의 프로세서(120), 컴퓨터 판독 가능한 저장매체(130) 및 통신 버스(170)를 포함한다.

[0026] 프로세서(120)는 지식 그래프 완성 장치(110)로 동작하도록 제어할 수 있다. 예컨대, 프로세서(120)는 컴퓨터

판독 가능한 저장 매체(130)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 컴퓨터 실행 가능 명령어는 프로세서(120)에 의해 실행되는 경우 지식 그래프 완성 장치(110)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0027] 컴퓨터 판독 가능한 저장 매체(130)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능한 저장 매체(130)에 저장된 프로그램(140)은 프로세서(120)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독한 가능 저장 매체(130)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 지식 그래프 완성 장치(110)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0028] 통신 버스(170)는 프로세서(120), 컴퓨터 판독 가능한 저장 매체(140)를 포함하여 지식 그래프 완성 장치(110)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0029] 지식 그래프 완성 장치(110)는 또한 하나 이상의 입출력 장치(24)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(150) 및 하나 이상의 통신 인터페이스(160)를 포함할 수 있다. 입출력 인터페이스(150) 및 통신 인터페이스(160)는 통신 버스(170)에 연결된다. 입출력 장치(미도시)는 입출력 인터페이스(150)를 통해 지식 그래프 완성 장치(110)의 다른 컴포넌트들에 연결될 수 있다.

[0030] 지식 그래프 완성 장치(110)는 기존 정보들을 기반으로 누락되거나 부정확한 엔티티 또는 관계에 대해서 멀티 홉 이웃 정보를 활용하여 엔티티 또는 관계을 추론하여 지식 그래프를 정제한다. 지식 그래프 완성 장치(110)는 주변 이웃뿐만 아니라 멀티 홉 이웃을 상황 정보로 활용한다. 지식 그래프 완성 장치(110)는 다양한 연결 패턴을 활용하면서 중요한 연결 패턴만을 추출하여 학습에 활용하기 위해 컨볼루션 연산을 모델에 포함시켜 학습을 수행한다.

[0031] 지식 그래프 추론 모델은 엔티티와 해당 엔티티의 멀티 홉 이웃 정보의 결합을 통해 학습을 수행하고 지식 그래프 내의 엔티티와 관계를 벡터화하는 모델이다. 지식 그래프 임베딩 과정은 엔티티 사이의 연결 패턴에 기반한 구조적 유사도에 따라 학습을 수행한다.

[0032] 도 3은 본 발명의 다른 실시예에 따른 지식 그래프 완성 방법을 예시한 흐름도이다. 지식 그래프 완성 방법은 컴퓨팅 디바이스에 의하여 수행될 수 있으며, 지식 그래프 완성 장치와 동일한 방식으로 동작한다.

[0033] 단계 S210에서 프로세서는 지식 그래프에서 멀티 홉 이웃 관계에 있는 문맥 정보를 수집한다. 문맥 정보는 엔티티를 포함하고 엔티티 간의 관계를 포함한다. 루트 엔티티의 멀티 홉 이웃에는 다양한 연결 패턴이 포함되어 있다. 지식 그래프 완성 방법은 멀티 홉 이웃 정보를 활용하여 연결 패턴으로부터 엔티티들의 구조적 역할 유사도를 파악한다.

[0034] 단계 S220에서 프로세서는 문맥 정보를 지식 그래프 학습 모델을 통하여 추론한다. 지식 그래프 완성 방법은 컨볼루션 연산을 활용하여 중요한 연결 패턴만을 추출하여 엔티티를 결합한다.

[0035] 이하에서는 도 4 내지 도 8을 참조하여, 문맥 정보를 수집하는 것을 설명하기로 한다.

[0036] 도 4는 본 발명의 다른 실시예에 따른 지식 그래프 완성 방법에서 문맥 정보를 수집하는 단계를 예시한 흐름도이고, 도 5 내지 도 8은 문맥 정보를 수집하는 단계를 예시한 도면이다.

[0037] 단계 S310에서 프로세서는 지식 그래프에 속하는 루트 엔티티로부터 기 설정된 윈도우 크기(k) 범위 내에서 너비 우선 검색(Breadth First Search)을 수행하여 지식 그래프(G)에 속하는 엔티티(e)를 획득하여 문맥 정보(Ce)에 해당하는 멀티 홉 이웃 정보(Ne)에 추가한다. 너비 우선 검색은 시작 엔티티부터 가까운 엔티티를 먼저 방문하고 멀리 떨어져 있는 엔티티를 나중에 방문하는 순회 방법이다. 도 5에 지식 그래프의 임의의 루트 엔티티에서 멀티 홉 이웃 관계에 있는 엔티티를 검색한 예시가 도시되어 있다.

[0038] 단계 S320에서 프로세서는 획득한 엔티티의 개수가 윈도우 크기 범위보다 작으면 다음 이웃 엔티티를 멀티 홉 이웃 정보에 추가한다.

[0039] 단계 S330에서 프로세서는 문맥 정보에 해당하는 멀티 홉 이웃 정보에서 엔티티 간의 거리를 기준으로 고유한 순위를 지정하고 기 설정된 윈도우 크기 범위 내에서 일부의 엔티티를 제거하여 상기 문맥 정보를 정규화한다.

[0040] 엔티티 간의 거리가 가까울수록 엔티티의 순위가 높아진다. 엔티티의 순위를 지정하는 방식은 수학식 1과 같이 표현된다.

수학식 1

[0041] $$\forall u, w \in N_e : dist(u,e) < dist(w,e) \Longrightarrow rank(u) < rank(w)$$

[0042] 수학식 1에서 dist()는 두 개의 엔티티의 거리이고, rank()는 하나의 엔티티의 순위이다.

[0043] 도 6에 멀티 홉 이웃 관계에 있는 엔티티를 정규화한 예시가 도시되어 있다. 예컨대, 윈도우 범위가 8이면 순위 9에 해당하는 엔티티와 순위 10에 해당하는 엔티티를 제외시킨다.

[0044] 단계 S340에서 프로세서는 문맥 정보에 해당하는 멀티 홉 이웃 정보의 그룹별 순위에 따라 깊이 우선 탐색(Depth First Search)을 수행하여 멀티 홉 이웃 정보를 비선형 공간으로부터 선형 공간으로 매핑하여 문맥 정보를 변환한다. 깊이 우선 탐색은 한 방향으로 갈 수 있을 때까지 계속 검색하다가 더 이상 갈 수 없게 되면 다시 가장 가까운 분기로 돌아와서 분기부터 다른 방향으로 다시 탐색을 진행하는 방법이다. 도 7에 멀티 홉 이웃 관계에 있는 엔티티를 우선 순위에 따라 일 방향으로 검색한 예시가 도시되어 있다.

[0045] 지식 그래프 완성 방법이 그래프 형태의 문맥 정보를 선형 공간으로 매핑하여 문맥 정보를 변환하는 이유는 지식 그래프 학습 모델의 입력 형태에 맞도록 변경할 필요가 있기 때문이다. 도 8을 참조하면 지식 그래프에서 추출된 멀티 홉 이웃 정보를 정규화하고 필터링한 후 1차원으로 변경된 멀티 홉 이웃 정보가 도시되어 있다.

[0046] 이하에서는 도 9 및 도 10을 참조하여, 문맥 정보를 추론하는 것을 설명하기로 한다.

[0047] 도 9는 본 발명의 다른 실시예에 따른 지식 그래프 완성 방법에서 문맥 정보를 추론하는 단계를 예시한 흐름도이고, 도 10은 지식 그래프 학습 모델을 예시한 도면이다.

[0048] 지식 그래프 완성 방법은 팩트의 유효성을 확인하기 위해 주어진 <h, r, t> 팩트에 점수를 매긴다. 후보 엔티티인 h 및 t 각각에 대해 점수화 함수로 <h, r, t> 팩트를 평가한다. 지식 그래프 완성 방법은 <h, r, ?> 또는 <?, r, t>와 같은 가능한 후보 엔티티를 예측한다.

[0049] 단계 S410에서 프로세서는 제1 레이어를 이용하여 엔티티와 관계에 관한 문맥 정보를 복수 차원의 임베딩 행렬에서 문맥 정보 임베딩으로 인코딩한다. 지식 그래프 학습 모델은 엔티티와 관계에 관한 문맥 정보를 복수 차원(d)의 임베딩 행렬에서 문맥 정보 임베딩으로 인코딩하는 제1 레이어를 포함한다. 임베딩은 데이터를 정해진 차원의 벡터에 대응시키는 것으로 모든 차원이 값을 갖는 벡터로 표현된다. 문맥 정보가 고정된 크기(s)를 갖도록 문맥 정보 임베딩에 가변적 크기를 갖는 패딩 벡터가 추가된다.

[0050] 단계 S420에서 프로세서는 제2 레이어를 이용하여 문맥 정보로부터 연결 패턴에 관한 문맥 특징 벡터를 추출한다. 지식 그래프 학습 모델은 컨볼루션 필터를 적용하여 문맥 정보로부터 연결 패턴에 관한 문맥 특징 벡터를 추출하는 제2 레이어를 포함한다. 제2 레이어는 특징을 추출하는 컨볼루션 레이어(Convolution Layer)와 대표 특징을 선택하여 서브 샘플링을 수행하는 맥스 풀링 레이어(Max Pooling Layer)를 포함할 수 있다.

[0051] 지식 그래프 학습 모델은 컨볼루션 필터를 공유할 수 있고, i번째 특징맵 fe는 수학식 2와 같이 표현된다.

수학식 2

[0052] $$\mathbf{f}_e^i = \{\sigma(\omega^i * \mathbf{C}_{j:j+h-1}^i + b^i) | 1 \le j \le s-h+1\}$$

[0053] 수학식 2에서 h는 필터의 높이이고, s는 고정된 크기이고, σ는 ReLu(Rectified Linear Unit) 등의 비선형 활성화 함수이다.

[0054] 레이어는 파라미터를 포함할 수 있고, 레이어의 파라미터는 학습가능한 필터 집합을 포함한다. 파라미터는 노드 간의 가중치(ω) 및/또는 바이어스(b)를 포함한다.

[0055] 단계 S430에서 프로세서는 제3 레이어를 이용하여 문맥 특징 벡터와 엔티티 임베딩을 벡터 공간에서 결합한다. 지식 그래프 학습 모델은 문맥 특징 벡터와 제1 레이어에 의해 인코딩된 엔티티 임베딩을 벡터 공간에서 결합하는(concatenate) 제3 레이어를 포함한다.

수학식 3

[0056]
$$\bar{\mathbf{c}}_e = \begin{bmatrix} \mathbf{c}_e \\ \mathbf{e} \end{bmatrix},$$
$$\mathbf{e}_c = \sigma(\mathbf{W}_{jm}\bar{\mathbf{c}}_e + \mathbf{b}_{jm})$$

[0057] W는 가중치 벡터이고, b는 바이어스 벡터이고, e_c는 임베딩이 결합된 결과 벡터이다.

[0058] 단계 S440에서 프로세서는 결합된 문맥 특징 벡터 및 엔티티 임베딩에 관계 임베딩을 가중합하여 합성 벡터를 생성한다. 지식 그래프 학습 모델은 결합된 문맥 특징 벡터 및 엔티티 임베딩에 제1 레이어에 의해 인코딩된 관계 임베딩을 가중합하여 합성 벡터를 생성한다. 임베딩을 가중합하는 방식은 수학식 4와 같이 표현된다.

수학식 4

[0059]
$$\mathbf{e}_c \circ \mathbf{r} = \sigma((\mathbf{w}_c^1 * \mathbf{e}_c) + (\mathbf{w}_r^1 * \mathbf{r}) + \mathbf{b}_{cr}^1), \ (e = h)$$
$$\mathbf{e}_c \circ \mathbf{r} = \sigma((\mathbf{w}_c^2 * \mathbf{e}_c) + (\mathbf{w}_r^2 * \mathbf{r}) + \mathbf{b}_{cr}^2), \ (e = t)$$

[0060] 팩트는 방향성을 갖기 때문에, (h, r)과 (t, r) 쌍에 대해 두 결합 상태가 고려할 필요가 있다.

[0061] 단계 S450에서 프로세서는 제4 레이어를 이용하여 합성 벡터에 후보 엔티티 임베딩을 행렬곱하여 점수를 산출한다. 지식 그래프 학습 모델은 합성 벡터에 제1 레이어에 의해 인코딩된 후보 엔티티 임베딩을 행렬곱하여 점수를 산출하는 제4 레이어를 포함한다.

[0062] (e, r) 쌍의 합성 벡터 간의 관계와 후보 엔티티 p의 임베딩은 <e, r, p> 또는 <p, r, e>의 유효성을 나타낸다.

[0063] 점수화 함수는 수학식 5과 같이 표현된다.

수학식 5

[0064]
$$\psi_p(C_e, e, r) = f(\mathbf{p}^T(\mathbf{e}_c \circ \mathbf{r}) + b_0)$$

[0065] 1-N 동시 예측 방식은 수학식 5에서 수학식 6으로 확장될 수 있다.

수학식 6

[0066]
$$\{\psi_{p_i}(C_e, e, r) | p_i \in \mathbf{P}\} = f(\mathbf{P}^T(\mathbf{e}_c \circ \mathbf{r}) + b_0)$$

[0067] 수학식 6은 후보 엔티티들 P에 대해 예측 점수의 확률 분포를 생성한다.

[0068] 지식 그래프 학습 모델은 소프트 맥스 손실 함수가 최소화하도록 학습한다. 지식 그래프 학습 모델에 네거티브 샘플링이 적용될 수 있다. 네거티브 샘플링은 등장했던(positive) 엔티티가 아닌 등장하지 않았던(negative) 엔티티를 이용하여 학습하는 방식이다.

[0069] 소프트 맥스 손실 함수는 수학식 7과 같이 표현된다.

수학식 7

$$\mathcal{L} = -\sum_{i}^{|P|} \frac{\mathbb{I}(y_i = 1)}{\sum_{j}^{|P|} \mathbb{I}(y_j = 1)} log(\psi_p(C_e, e, r)_i)$$

[0070]

[0071] yi=1은 후보 엔티티 세트 P의 엔티티 i가 참(true)인 것을 나타낸다. $\mathbb{I}$는 지시 함수이며, 엔티티 i가 참이면 1이고, 엔티티 i가 거짓이면 0을 나타낸다.

[0072] 지식 그래프 완성 장치는 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합에 의해 로직회로 내에서 구현될 수 있고, 범용 또는 특정 목적 컴퓨터를 이용하여 구현될 수도 있다. 장치는 고정배선형(Hardwired) 기기, 필드 프로그램 가능한 게이트 어레이(Field Programmable Gate Array, FPGA), 주문형 반도체(Application Specific Integrated Circuit, ASIC) 등을 이용하여 구현될 수 있다. 또한, 장치는 하나 이상의 프로세서 및 컨트롤러를 포함한 시스템온칩(System on Chip, SoC)으로 구현될 수 있다.

[0073] 지식 그래프 완성 장치는 하드웨어적 요소가 마련된 컴퓨팅 디바이스 또는 서버에 소프트웨어, 하드웨어, 또는 이들의 조합하는 형태로 탑재될 수 있다. 컴퓨팅 디바이스 또는 서버는 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통신 모뎀 등의 통신장치, 프로그램을 실행하기 위한 데이터를 저장하는 메모리, 프로그램을 실행하여 연산 및 명령하기 위한 마이크로프로세서 등을 전부 또는 일부 포함한 다양한 장치를 의미할 수 있다.

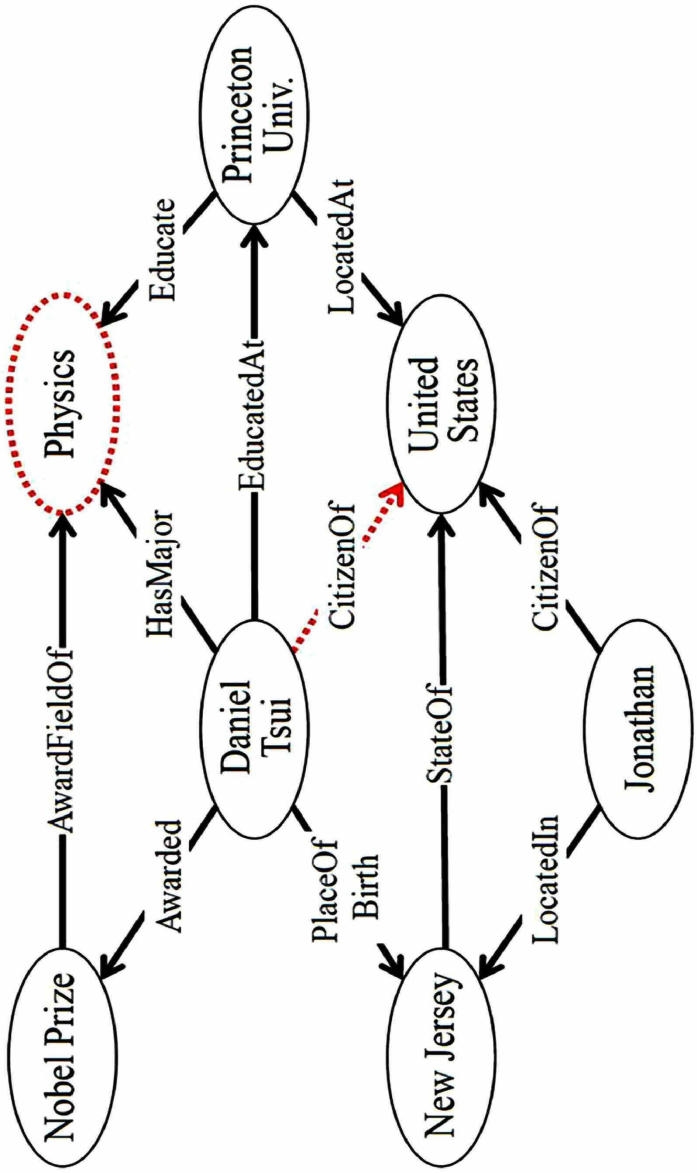
[0074] 도 3, 도 4, 도 8에서는 각각의 과정을 순차적으로 실행하는 것으로 기재하고 있으나 이는 예시적으로 설명한 것에 불과하고, 이 분야의 기술자라면 본 발명의 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도 3, 도 4, 도 8에 기재된 순서를 변경하여 실행하거나 또는 하나 이상의 과정을 병렬적으로 실행하거나 다른 과정을 추가하는 것으로 다양하게 수정 및 변형하여 적용 가능할 것이다.

[0075] 본 실시예들에 따른 동작은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능한 매체에 기록될 수 있다. 컴퓨터 판독 가능한 매체는 실행을 위해 프로세서에 명령어를 제공하는데 참여한 임의의 매체를 나타낸다. 컴퓨터 판독 가능한 매체는 프로그램 명령, 데이터 파일, 데이터 구조 또는 이들의 조합을 포함할 수 있다. 예를 들면, 자기 매체, 광기록 매체, 메모리 등이 있을 수 있다. 컴퓨터 프로그램은 네트워크로 연결된 컴퓨터 시스템 상에 분산되어 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다. 본 실시예를 구현하기 위한 기능적인(Functional) 프로그램, 코드, 및 코드 세그먼트들은 본 실시예가 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있을 것이다.

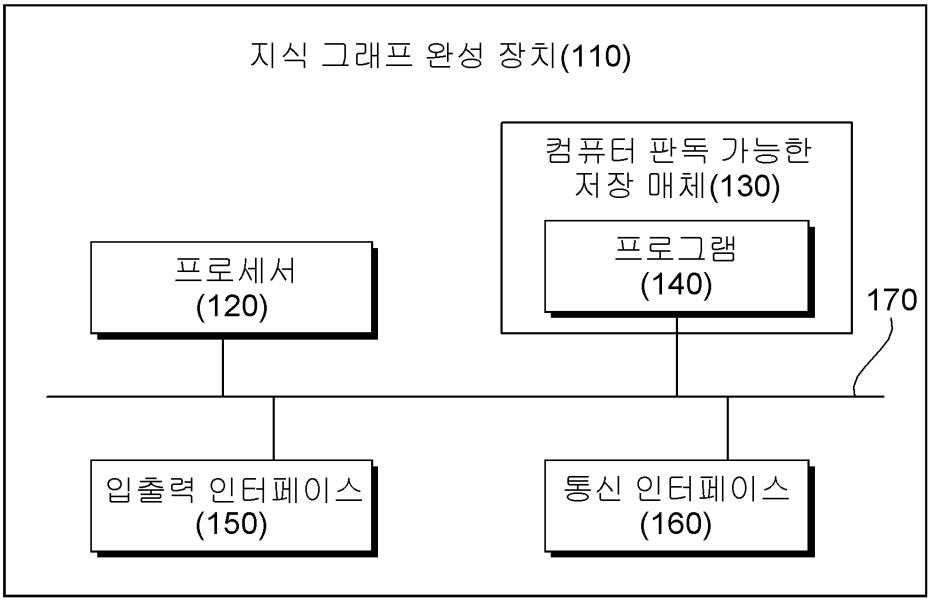
[0076] 본 실시예들은 본 실시예의 기술 사상을 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상의 범위가 한정되는 것은 아니다. 본 실시예의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면

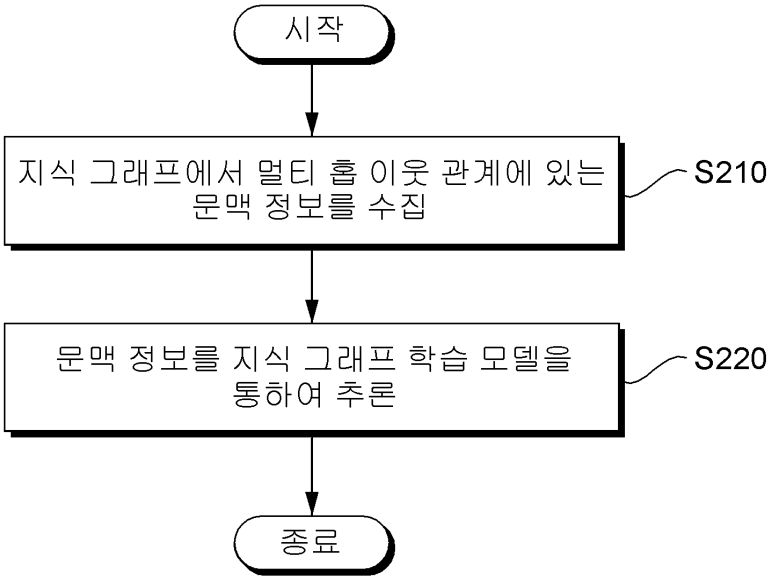
도면1


Princeton Univ.
Educate
LocatedAt
Physics
United States
EducatedAt
HasMajor
CitizenOf
CitizenOf
AwardFieldOf
Daniel Tsui
StateOf
Jonathan
Awarded
PlaceOf Birth
LocatedIn
Nobel Prize
New Jersey

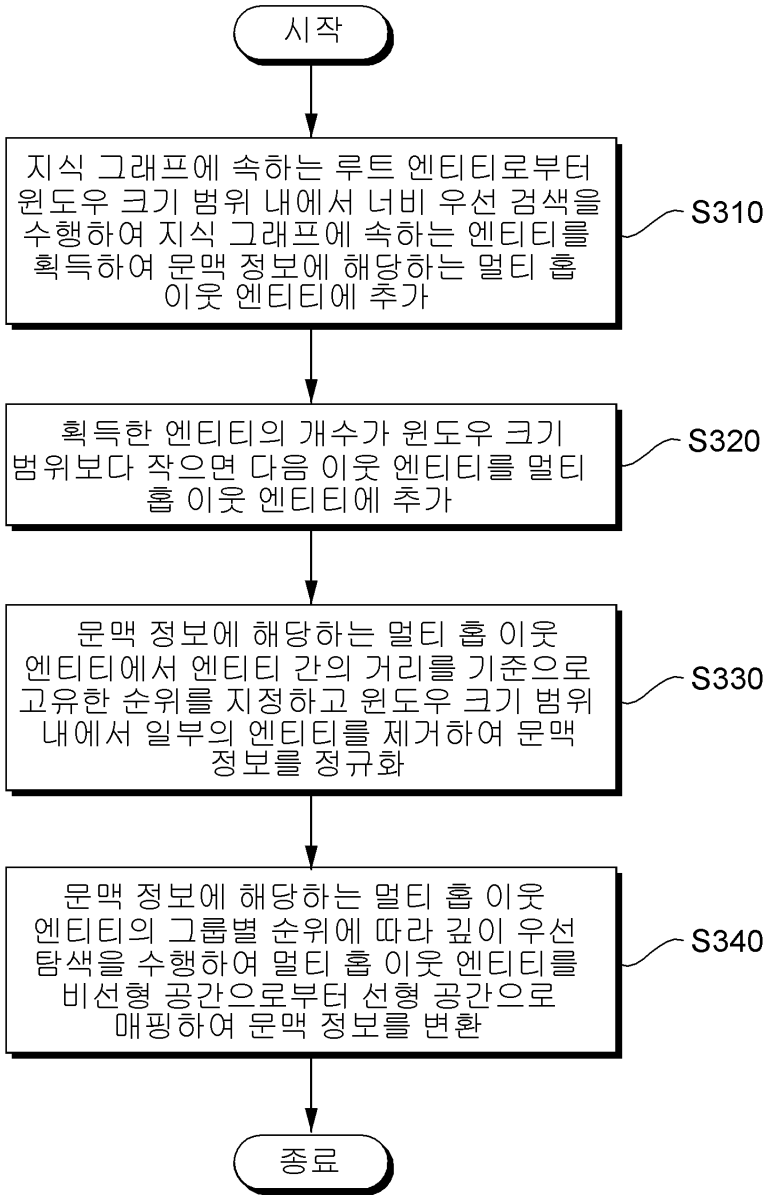
도면2


지식 그래프 완성 장치(110)
컴퓨터 판독 가능한
저장 매체(130)
프로그램
(140)
프로세서
(120)
170
입출력 인터페이스
(150)
통신 인터페이스
(160)

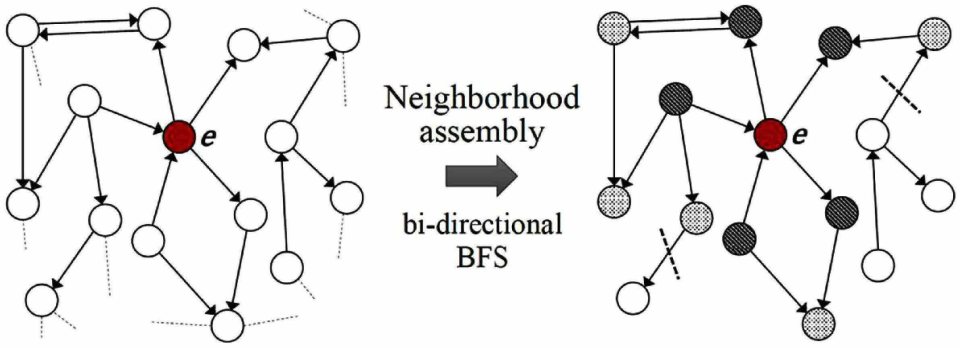
도면3


시작
지식 그래프에서 멀티 홉 이웃 관계에 있는
문맥 정보를 수집
S210
문맥 정보를 지식 그래프 학습 모델을
통하여 추론
S220
종료

도면4

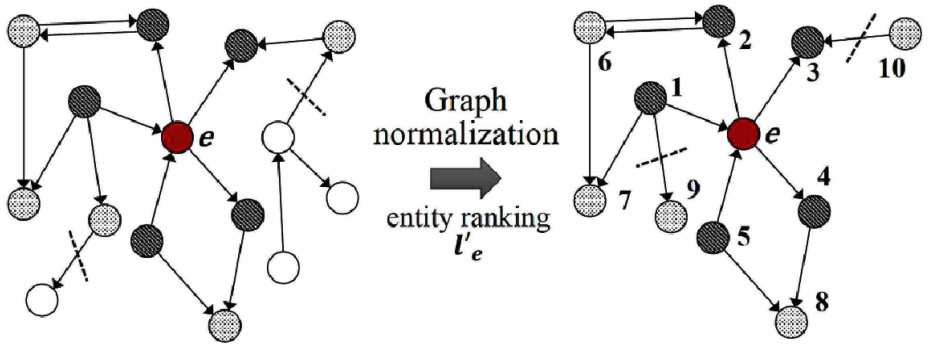

시작
지식 그래프에 속하는 루트 엔티티로부터 윈도우 크기 범위 내에서 너비 우선 검색을 수행하여 지식 그래프에 속하는 엔티티를 획득하여 문맥 정보에 해당하는 멀티 홉 이웃 엔티티에 추가
S310
획득한 엔티티의 개수가 윈도우 크기 범위보다 작으면 다음 이웃 엔티티를 멀티 홉 이웃 엔티티에 추가
S320
문맥 정보에 해당하는 멀티 홉 이웃 엔티티에서 엔티티 간의 거리를 기준으로 고유한 순위를 지정하고 윈도우 크기 범위 내에서 일부의 엔티티를 제거하여 문맥 정보를 정규화
S330
문맥 정보에 해당하는 멀티 홉 이웃 엔티티의 그룹별 순위에 따라 깊이 우선 탐색을 수행하여 멀티 홉 이웃 엔티티를 비선형 공간으로부터 선형 공간으로 매핑하여 문맥 정보를 변환
S340
종료

도면5



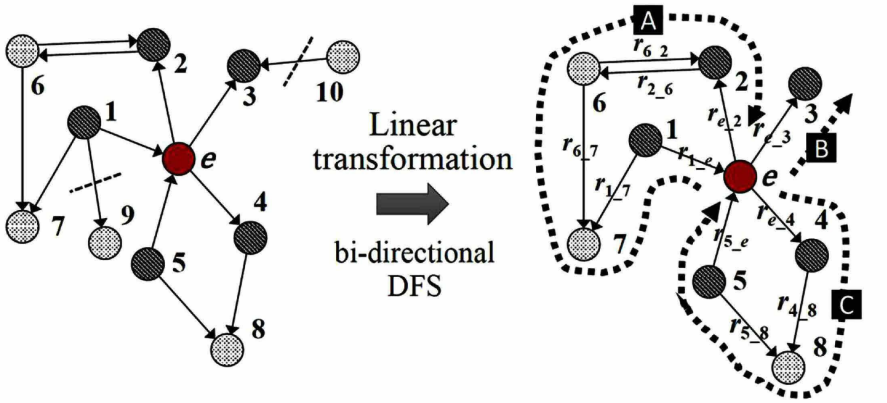
(a) *knowledge graph* $(\mathcal{G})$ **(b)** *multi-hop neighborhood* (N_e)

도면6



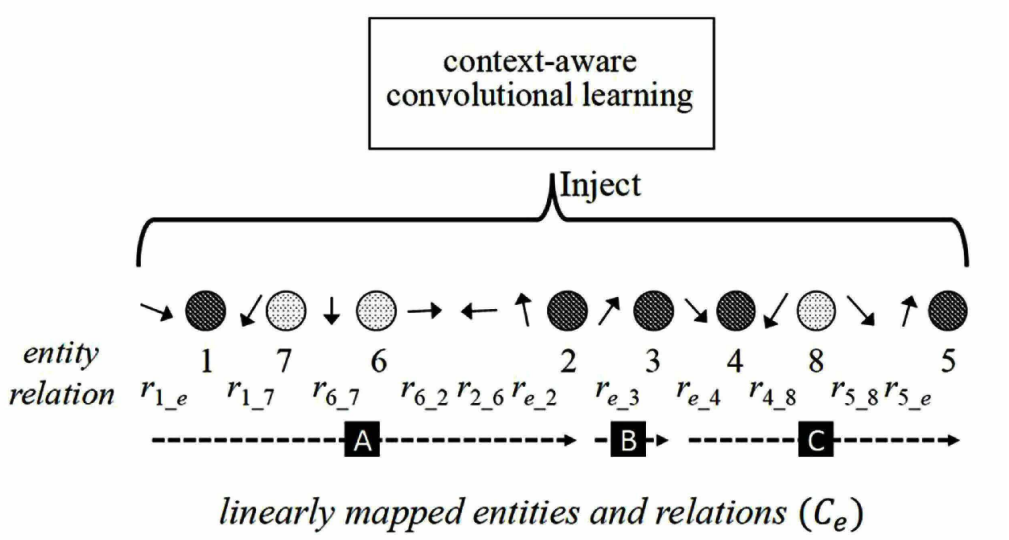
(b) *multi-hop neighborhood* (N_e) **(c)** *normalized multi-hop neighborhood* $(N_{e,n})$

도면7

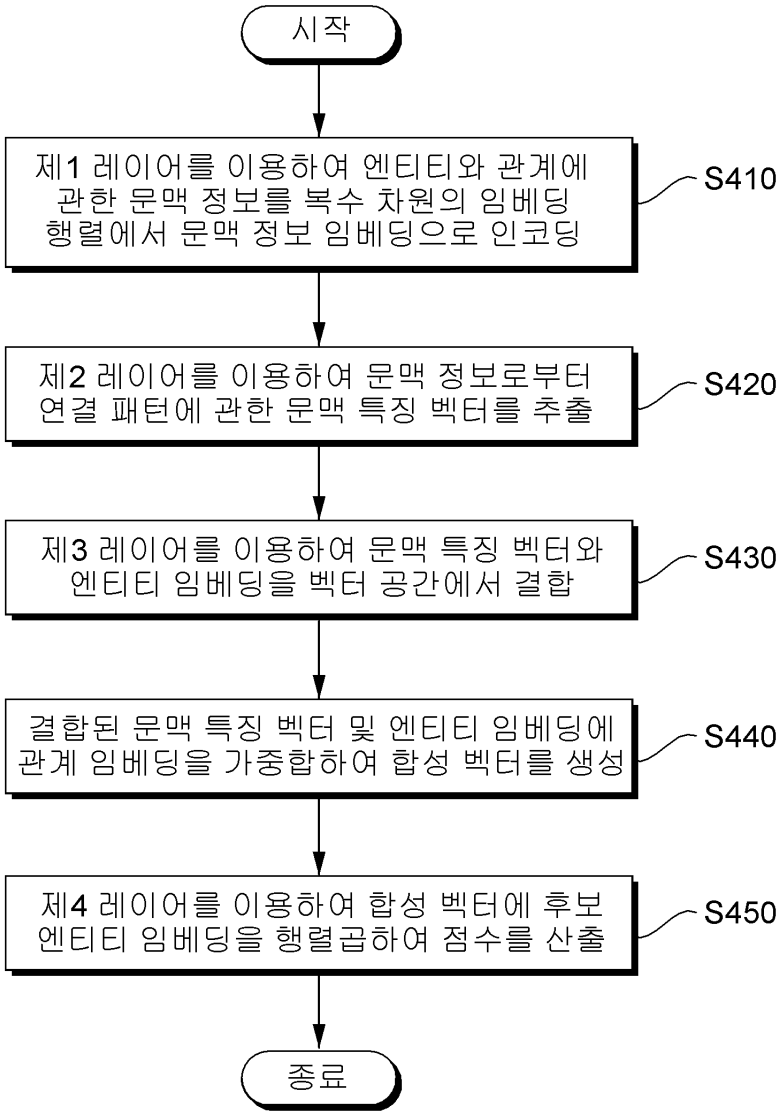


(c) *normalized multi-hop neighborhood* $(N_{e,n})$ **(d)** *linearly mapped entities and relations* (C_e)

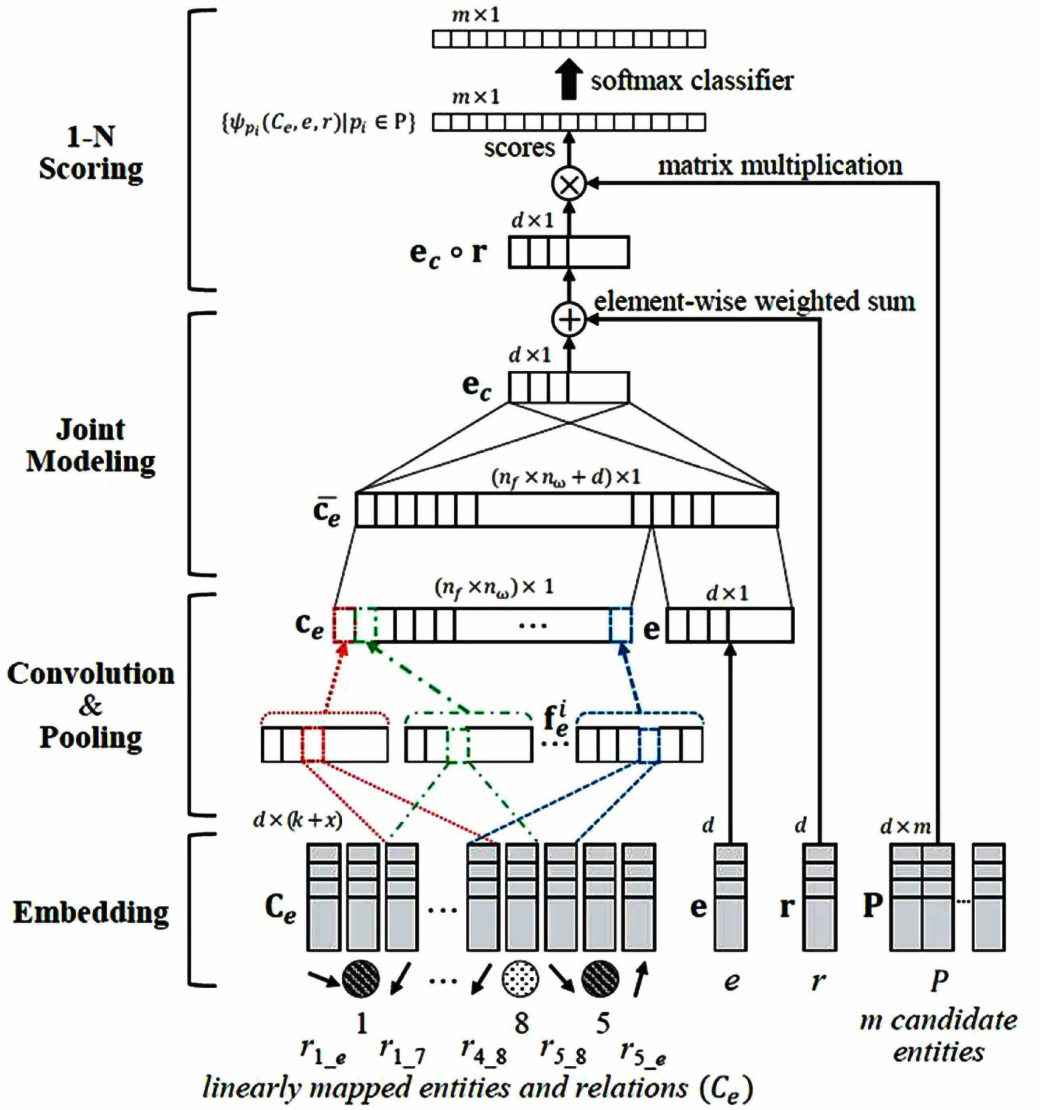
도면8


context-aware convolutional learning
Inject
entity
relation
1 7 6 2 3 4 8 5
r_{1_e} r_{1_7} r_{6_7} r_{6_2} r_{2_6} r_{e_2} r_{e_3} r_{e_4} r_{4_8} r_{5_8} r_{5_e}
A
B
C
linearly mapped entities and relations (C_e)

도면9


시작
제1 레이어를 이용하여 엔티티와 관계에 관한 문맥 정보를 복수 차원의 임베딩 행렬에서 문맥 정보 임베딩으로 인코딩
S410
제2 레이어를 이용하여 문맥 정보로부터 연결 패턴에 관한 문맥 특징 벡터를 추출
S420
제3 레이어를 이용하여 문맥 특징 벡터와 엔티티 임베딩을 벡터 공간에서 결합
S430
결합된 문맥 특징 벡터 및 엔티티 임베딩에 관계 임베딩을 가중합하여 합성 벡터를 생성
S440
제4 레이어를 이용하여 합성 벡터에 후보 엔티티 임베딩을 행렬곱하여 점수를 산출
S450
종료

도면10


1-N Scoring
Joint Modeling
Convolution & Pooling
Embedding
m×1
softmax classifier
m×1
{ψ_pi(C_e, e, r)|p_i ∈ P}
scores
matrix multiplication
d×1
e_c ∘ r
element-wise weighted sum
d×1
e_c
(n_f × n_ω + d)×1
c̄_e
(n_f × n_ω)× 1
d×1
c_e
e
f_e^i
d×(k+x)
d
d
d×m
C_e
e
r
P
1
8
5
r_1_e r_1_7 r_4_8 r_5_8 r_5_e
linearly mapped entities and relations (C_e)
m candidate entities