(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년07월14일
(11) 등록번호 10-2277787
(24) 등록일자 2021년07월09일

(51) 국제특허분류(Int. Cl.)
G06F 16/2452 (2019.01) *G06F 16/2453* (2019.01)
G06F 16/2455 (2019.01) *G06F 3/08* (2006.01)
(52) CPC특허분류
G06F 16/24522 (2019.01)
G06F 16/24542 (2019.01)
(21) 출원번호 10-2019-0174966
(22) 출원일자 2019년12월26일
심사청구일자 2019년12월26일
(65) 공개번호 10-2021-0082726
(43) 공개일자 2021년07월06일
(56) 선행기술조사문헌
WO2018213530 A2*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
(72) 발명자
한욱신
경상북도 포항시 남구 청암로 77 창의IT융합공학과 (지곡동, 포항공과대학교)
나인혁
서울특별시 서대문구 연희로32길 48, 104동 105호(연희동, 연희동성원아파트)
(뒷면에 계속)
(74) 대리인
특허법인이룸리온

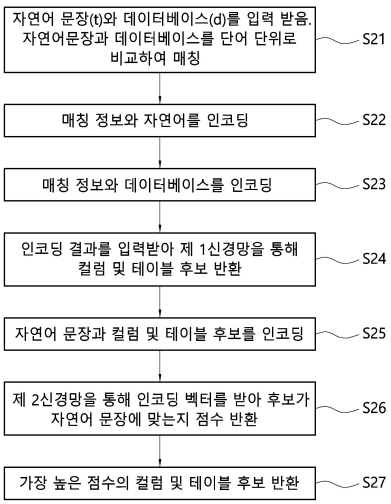
전체 청구항 수 : 총 3 항 심사관 : 이현중

(54) 발명의 명칭 **신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법**

(57) 요 약

본 발명은 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법에 관한 것으로, 마이크로컴퓨터를 포함하는 컴퓨팅 장치를 이용하여 자연어 문장과 데이터베이스 스키마가 주어졌을 때, 해당하는 SQL 질의에 사용될 컬럼 및 테이블을 예측하는 방법으로서, a) 제1신경망을 통해 SQL 질의에 사용될 컬럼 및 테이블의 후보들을 예측하는 단계와, b) 제2신경망을 통해 상기 컬럼 및 테이블의 후보들과 자연어 문장과의 일치도를 구하고, 일치도가 가장 높은 컬럼 및 테이블을 선택하는 단계를 포함한다.

대 표 도 - 도2



(52) CPC특허분류
G06F 16/24547 (2019.01)
G06F 16/24557 (2019.01)
G06F 3/08 (2020.08)

(72) 발명자
김현지
울산광역시 동구 월봉12길 50, C동 308호 (화정동, 송정타워맨션3차)

강혁규
경상북도 포항시 남구 효자동길5번길 23, 201호(효자동)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711082907
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기술진흥센터
연구사업명	SW컴퓨팅산업원천기술개발
연구과제명	대화 가능하고 자동으로 튜닝하는 DBMS의 개발
기 여 율	1/1
과제수행기관명	포항공과대학교 산학협력단
연구기간	2019.02.01 ~ 2019.12.31

명 세 서

청구범위

청구항 1

마이크로컴퓨터를 포함하는 컴퓨팅 장치를 이용하여 자연어 문장과 데이터베이스 스키마가 주어졌을 때, 해당하는 SQL 질의에 사용될 컬럼 및 테이블을 예측하는 방법으로서,

a-1) 자연어 문장에 나타나는 단어들과 데이터베이스 스키마의 컬럼 및 테이블을 연결하되, 자연어 문장과 데이터베이스를 단어 단위로 비교하여 매칭하는 과정;

a-2) 상기 a-1) 과정의 결과를 기반으로 자연어 및 스키마를 제1신경망을 사용하여 인코딩하여 각 컬럼에 해당하는 임베딩 벡터들 및 각 테이블에 해당하는 임베딩 벡터들을 구하되, 사전 학습된 트랜스포머 신경망을 통해 각 단어, 컬럼의 단어, 테이블들의 단어에 대한 임베딩 벡터를 구한 후, 동일 컬럼 또는 동일 테이블에 속하는 단어들끼리는 임베딩 벡터들끼리의 합 연산을 통해 컬럼 및 테이블에 해당하는 벡터를 구하는 과정;

a-3) 상기 a-2) 과정에서 인코딩된 자연어 문장 및 스키마를 입력으로 받아 순차적으로 SQL 질의에 사용될 컬럼 및 테이블들을 반환하는 과정; 및

a-4) 빔 서치를 통해 SQL 질의에 사용될 컬럼 및 테이블의 후보들을 생성하는 과정으로 이루어지는 a) 단계; 및

b-1) 자연어 문장과 컬럼 및 테이블의 후보 이름을 사전 학습된 제2신경망을 이용하여 유사도를 구하는 과정; 및

b-2) 상기 컬럼 및 테이블 후보 중 가장 유사도가 높은 컬럼 및 테이블을 선택하는 과정으로 이루어지는 b) 단계를 포함하는 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 제2신경망은 트랜스포머 신경망이며,

데이터베이스 스카마로부터 참인 후보 및 거짓인 후보들을 추출하여 학습하는 것을 특징으로 하는 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법.

청구항 6

제5항에 있어서,

상기 제2신경망은,

다중 퍼셉트론 신경망을 적용하여 시그모이드(sigmoid) 함수를 적용해 유사도를 0부터 1 사이의 실수값으로 반환하는 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 자연어로부터 SQL 질의 번역시 컬럼 및 테이블을 예측하는 방법에 관한 것으로, 더 상세하게는 자연어 데이터베이스 스키마를 인코딩하여 후보 컬럼 및 테이블 집합을 만들 때 가장 확률이 높은 컬럼 및 테이블 집합을 예측하는 방법에 관한 것이다.

배 경 기 술

[0002] 일반적으로, 자연어를 SQL 질의로 번역하는 문제는 전문 지식이 없는 사람이 관계형 데이터베이스에 질의할 수 있다는 점에서 중요한 문제이다. 최근 SQL 질의 번역 문제를 푸는 많은 규칙 및 신경망 기반 방법들이 제안되었다.

[0003] 하지만, 규칙 방법은 사람이 직접 매핑 사전을 만들어 주어야 하는 문제가 있으며, 신경망 기반 방법들은 정확도가 낮은 문제가 있다. 정확도가 낮은 이유는 특히 SQL 질의에 사용되는 컬럼 및 테이블을 잘 예측하지 못하기 때문에, 컬럼 및 테이블을 정확하게 예측하는 방법이 요구되고 있다.

[0004] "Diptikalyan Saha, Avrilia Floratou, Karthik Sankaranarayanan, Umar Farooq Minhas, Ashish R Mittal, Fatma Özcan : ATHENA: An Ontology-Driven System for Natural Language Querying over Relational Data Stores PVLDB 9(12): 1209-1220 (2016)(선행문헌1)"에서는 자연어를 SQL 질의로 변환하는 규칙 기반 기술 ATHENA를 제안하였다. ATHENA는 전문가가 사전 정의한 온톨로지(ontology) 및 텍스트의 단어와 온톨로지의 개체를 매핑하는 사전을 이용해 자연어의 단어들을 온톨로지의 개체들에 매핑한다. 이렇게 매핑된 개체들 및 자연어를 이용해 OQL(Ontology Query Language) 질의를 생성하고, 다시 이를 SQL 질의로 변환하는 방법으로 SQL 질의 번역을 수행한다.

[0005] 이러한 방식은 온톨로지 및 사전을 전문가가 미리 만들어 주어야만 동작하는 문제가 있으며, 데이터베이스 스키마가 바뀌면 온톨로지 및 사전을 수정해야 하는 문제가 있다.

[0006] 신경망 기술의 발전과 SQL 질의 번역 데이터가 발표되어 신경망 기반 SQL 질의 번역 기술이 활발히 연구되고 있다. 최근의 신경망 기반 연구들은 크게 슬롯 채우기(slot filling)와 문맥 자유 문법 기반 방식으로 나눌 수 있다.

[0007] 슬롯 채우기 방법은 특정 템플릿을 갖는 단순한 SQL 질의를 가정하여 템플릿의 각 슬롯을 신경망을 통해 채우는 방식으로, "Victor Zhong, Caiming Xiong, Richard Socher: Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning CoRR abs/170900103 (2017)(선행문헌2)"와 "Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, Dragomir R Radev : TypeSQL: Knowledge-Based Type-Aware Neural Text-to-SQL Generation NAACL-HLT (2) 2018 : 588-594(선행문헌3)"이 있다.

[0008] 선행문헌2는 특정 템플릿을 갖는 단순한 SQL 질의와 자연어 및 데이터베이스의 세 쌍으로 이루어진 데이터셋인 WikiSQL을 발표하였다.

[0009] 선행문헌 3은 WikiSQL에서의 SQL 질의 번역을 처음으로 슬롯 채우기 문제로 보아 TypeSQL이라는 신경망 기반 방법을 제시했다.

[0010] 하지만, WikiSQL에서와 같이 슬롯 채우기 방식을 적용할 수 있는 SQL 질의는 매우 간단하여 현실적으로 사용되기에는 무리가 있다. 예를 들어, 슬롯 채우기 방식들은 다양한 키워드(ORDER BY, GROUP BY 등)를 가정하지 않고, 중첩 질의를 처리할 수 없다.

[0011] 또한, 문맥 자유 문법 기반 방식의 예로는 "Wonseok Hwang, Jinyeung Yim, Seunghyun Park, Minjoon Seo: A Comprehensive Exploration on WikiSQL with Table-Aware Word Contextualization CoRR abs/190201069 (2019)(선행문헌4)"와 "Pengcheng He, Yi Mao, Kaushik Chakrabarti, Weizhu Chen : X-SQL: reinforce schema representation with context CoRR abs/190808113 (2019)(선행문헌5)"가 있다.

[0012] 위의 선행문헌4 및 선행문헌5는 많은 자연어로 사전 학습된 신경망에서의 전이 학습을 이용하는 SQLova 및 X-SQL을 제안하여 WikiSQL에서 약 90%의 높은 정확도를 얻었다.

[0013] 문맥 자유 문법 기반 방식은 SQL 질의와 같은 프로그래밍 언어의 문맥 자유 문법을 기반으로 신경망을 통해 최좌단 유도(leftmost derivation)하는 방식으로 최종 문장을 생성해 내는 방식으로, 미리 정의한 문맥 자유 문법

이 생성할 수 있는 모든 SQL 질의를 생성할 수 있어 슬롯 채우기 방식의 한계를 극복할 수 있다.

[0014] "Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, Dragomir R Radev : SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task EMNLP 2018 : 1653-1663(선행문헌6)"은 문맥 자유 문법 기반 방식으로 SQL 질의 번역을 하는 첫 연구이다.

[0015] 또한, "Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, Dongmei Zhang : Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation ACL (1) 2019 : 4524-4535(선행문헌7)"는 규칙 기반으로 데이터베이스 스키마와 자연어의 단어들을 연결하는 방법을 제안하였고, 또한 중간 언어인 SemQL을 제안하여 자연어에서 SemQL 질의로 번역하는 신경망 방식과 SemQL 질의를 SQL 질의로 번역하는 규칙 기반 방법을 통해 SQL 질의 번역을 수행한다.

[0016] 그러나 앞서 언급한 바와 같이 신경망 기반 방법들은 정확도가 낮은 문제가 있으며, 이에 대한 개선이 필요하다.

[0017] 도 1은 일반적인 관계형 데이터베이스에 대한 자연어 처리 흐름도이다.

[0018] 도 1을 참조하면 (a)와 같이 자연어 질의의 입력에 따라 (b)와 같이 SQL 질의로 변환하고, 그 변환된 내용을 (c)와 같이 관계형 데이터베이스에서 검색하여 그 결과를 자연어로 출력하게 된다.

[0019] 데이터 베이스 시스템의 자연어 인터페이스를 만드는 문제는 비전문가들이 데이터베이스에 쉽게 질의 할 수 있도록 한다는 점에서 중요한 문제다. 데이터베이스 시스템 중 가장 널리 쓰이는 관계형 데이터베이스에 대한 질의는 SQL 프로그래밍 언어로 이루어진다.

[0020] 따라서 자연어 질의를 SQL 질의로 번역함으로써 관계형 데이터베이스에 대한 자연어 인터페이스를 만들 수 있다.

[0021] 자연어 질의를 SQL 질의로 번역하는 문제의 큰 어려움 중 하나는 자연어 질의가 데이터베이스의 어떤 컬럼 및 테이블을 참조하는지 알아내는 것이다. 이것이 어려운 이유는 데이터베이스의 컬럼 및 테이블 이름이 자연어 질의에 그대로 등장하는 것이 아니라 의역되어 나타나거나 직접적으로 나타나지 않을 수 있기 때문이다. 또한, 데이터베이스 스키마에 비슷한 이름을 가진 컬럼 및 테이블들이 여러 개가 존재할 수 있어 정확하게 SQL 질의에 쓰일 컬럼 및 테이블들을 찾아내는 것은 어려운 문제이다.

[0022] 기존의 규칙 기반 SQL 질의 번역 방법들은 전문가가 사전에 매핑 사전을 만들어 주는 방식을 사용했지만, 이는 사전에 고려되지 않은 단어가 등장하거나, 데이터베이스 스키마가 달라지는 상황 등에서 문제가 생긴다. 또한, 모든 데이터베이스마다 전문가의 큰 노력이 드는 근본적인 문제가 있다.

[0023] 따라서 최근에는 신경망 기반 SQL 질의 번역 기술이 연구되고 있다. 신경망 기반 방법은 기존 규칙 기반 기술과는 달리 사용되는 모든 데이터베이스마다 전문가의 노력이 필요하지는 않다. 하지만 복잡한 질의에서 정확도가 낮은 문제가 있는데, 자연어 질의가 데이터베이스의 어떤 컬럼 및 테이블을 참조하는지 알아내는 정확도가 낮기 때문이다. IRNet은 이 문제를 지적하여 규칙 기반 방법으로 데이터베이스 스키마와 자연어의 단어들을 연결하는 기법을 제안하였지만, 규칙에 적용되지 않는 상황들이 많아 문제를 완전히 해결하지 못한다.

[0024]

발명의 내용

해결하려는 과제

[0025] 본 발명이 해결하고자 하는 기술적 과제는, 신경망 기반 방법들에서 컬럼 및 테이블을 정확하게 예측하여, 정확도를 높일 수 있는 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법을 제공함에 있다.

[0026] 좀 더 구체적으로, 입력된 자연어와 데이터베이스 스키마를 인코딩하여 후보 컬럼 및 테이블 집합을 만들어내는 제1신경망과, 제1신경망과는 별도로 제1신경망이 만들어낸 후보 컬럼 및 테이블 집합에서 가장 확률이 높은 컬럼 및 테이블을 선택하는 제2신경망을 이용하여 컬럼 및 테이블을 예측하는 방법을 제공함에 있다.

과제의 해결 수단

[0027] 상기와 같은 과제를 해결하기 위한 본 발명 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이

블을 예측하는 방법은, 마이크로컴퓨터를 포함하는 컴퓨팅 장치를 이용하여 자연어 문장과 데이터베이스 스키마가 주어졌을 때, 해당하는 SQL 질의에 사용될 컬럼 및 테이블을 예측하는 방법으로서, a) 제1신경망을 통해 SQL 질의에 사용될 컬럼 및 테이블의 후보들을 예측하는 단계와, b) 제2신경망을 통해 상기 컬럼 및 테이블의 후보들과 자연어 문장과의 일치도를 구하고, 일치도가 가장 높은 컬럼 및 테이블을 선택하는 단계를 포함한다.

[0028] 본 발명의 실시예에서, 상기 a) 단계는 a-1) 자연어 문장에 나타나는 단어들과 데이터베이스 스키마의 컬럼 및 테이블을 연결하는 과정과, a-2) 상기 a-1) 과정의 결과를 기반으로 자연어 및 스키마를 제1신경망을 사용하여 인코딩하는 과정과, a-3) 상기 a-2) 과정에서 인코딩된 자연어 문장 및 스키마를 입력으로 받아 순차적으로 SQL 질의에 사용될 컬럼 및 테이블들을 반환하는 과정과, a-4) 빔 서치를 통해 SQL 질의에 사용될 컬럼 및 테이블의 후보들을 생성하는 과정을 포함할 수 있다.

[0029] 본 발명의 실시예에서, 상기 a-1) 과정은 자연어 문장의 단어들과 컬럼 및 테이블의 이름을 연결하고, 자연어 문장의 단어들과 데이터베이스에 저장된 값을 비교하여 컬럼과 연결할 수 있다.

[0030] 본 발명의 실시예에서, 상기 b) 단계는 b-1) 자연어 문장과 컬럼 및 테이블의 후보 이름을 사전 학습된 제2신경망을 이용하여 유사도를 구하는 과정과, b-2) 상기 컬럼 및 테이블 후보 중 가장 유사도가 높은 컬럼 및 테이블을 선택하는 과정을 포함할 수 있다.

[0031] 본 발명의 실시예에서, 상기 제2신경망은 트랜스포머 신경망이며, 데이터베이스 스카마로부터 참인 후보 및 거짓인 후보들을 추출하여 학습할 수 있다.

[0032] 본 발명의 실시예에서, 상기 제2신경망은 다중 퍼셉트론 신경망을 적용하여 시그모이드(sigmoid) 함수를 적용해 유사도를 0부터 1 사이의 실수값으로 반환할 수 있다.

발명의 효과

[0033] 본 발명 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법은, 입력된 자연어와 데이터베이스 스키마를 인코딩하여 후보 컬럼 및 테이블 집합을 만들어내는 제1신경망과, 제1신경망과는 별도로 제1신경망이 만들어낸 후보 컬럼 및 테이블 집합에서 가장 확률이 높은 컬럼 및 테이블을 선택하는 제2신경망을 이용하여 가장 확률이 높은 컬럼 및 테이블 집합을 예측함으로써, 정확도를 높일 수 있는 효과가 있다.

도면의 간단한 설명

[0034] 도 1은 일반적인 관계형 데이터베이스에 대한 자연어 처리 흐름도이다.

도 2는 본 발명 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법의 순서도이다.

도 3은 도 2의 알고리즘이다.

도 4는 임베딩 벡터를 구하는 예시도이다.

발명을 실시하기 위한 구체적인 내용

[0035] 이하, 본 발명 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법에 대하여 첨부한 도면을 참조하여 상세히 설명한다.

[0036] 본 발명의 실시 예들은 당해 기술 분야에서 통상의 지식을 가진 자에게 본 발명을 더욱 완전하게 설명하기 위해 제공되는 것이며, 아래에 설명되는 실시 예들은 여러 가지 다른 형태로 변형될 수 있으며, 본 발명의 범위가 아래의 실시 예들로 한정되는 것은 아니다. 오히려, 이들 실시 예는 본 발명을 더욱 충실하고 완전하게 하며 당업자에게 본 발명의 사상을 완전하게 전달하기 위하여 제공되는 것이다.

[0037] 본 명세서에서 사용된 용어는 특정 실시 예를 설명하기 위하여 사용되며, 본 발명을 제한하기 위한 것이 아니다. 본 명세서에서 사용된 바와 같이 단수 형태는 문맥상 다른 경우를 분명히 지적하는 것이 아니라면, 복수의 형태를 포함할 수 있다. 또한, 본 명세서에서 사용되는 경우 "포함한다(comprise)" 및/또는"포함하는(comprising)"은 언급한 형상들, 숫자, 단계, 동작, 부재, 요소 및/또는 이들 그룹의 존재를 특정하는 것이며, 하나 이상의 다른 형상, 숫자, 동작, 부재, 요소 및/또는 그룹들의 존재 또는 부가를 배제하는 것이 아니다. 본 명세서에서 사용된 바와 같이, 용어 "및/또는"은 해당 열거된 항목 중 어느 하나 및 하나 이상의 모든 조합을

포함한다.

[0038] 본 명세서에서 제1, 제2 등의 용어가 다양한 부재, 영역 및/또는 부위들을 설명하기 위하여 사용되지만, 이들 부재, 부품, 영역, 층들 및/또는 부위들은 이들 용어에 의해 한정되지 않음은 자명하다. 이들 용어는 특정 순서나 상하, 또는 우열을 의미하지 않으며, 하나의 부재, 영역 또는 부위를 다른 부재, 영역 또는 부위와 구별하기 위하여만 사용된다. 따라서, 이하 상술할 제1 부재, 영역 또는 부위는 본 발명의 가르침으로부터 벗어나지 않고서도 제2 부재, 영역 또는 부위를 지칭할 수 있다.

[0039] 이하, 본 발명의 실시 예들은 본 발명의 실시 예들을 개략적으로 도시하는 도면들을 참조하여 설명한다. 도면들에서, 예를 들면, 제조 기술 및/또는 공차에 따라, 도시된 형상의 변형들이 예상될 수 있다. 따라서, 본 발명의 실시 예는 본 명세서에 도시된 영역의 특정 형상에 제한된 것으로 해석되어서는 아니 되며, 예를 들면 제조상 초래되는 형상의 변화를 포함하여야 한다.

[0040] 또한, 본 발명은 자연어를 SQL 질의로 번역(변환)하는 방법에서 컬럼 및 테이블을 예측하는 방법에 관한 것으로, 이러한 방법의 동작 주체는 통상의 연산이 가능한 제어기와 저장장치 및 데이터의 임시 저장 가능한 메모리를 포함하는 컴퓨팅 장치이며, 예를 들어 퍼스널 컴퓨터 또는 서버를 사용할 수 있다.

[0041] 도 2는 본 발명 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법의 순서도이고, 도 3은 도 2의 알고리즘이다.

[0042] 도 2와 도 3을 각각 참조하면 본 발명 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법은, 자연어 문장(t)과 데이터베이스(d)를 입력받아, 자연어 문장과 데이터베이스를 단어 단위로 비교하여 매칭하는 단계(S21, 1 : matching_info)와, 매칭 정보와 자연어 문장을 인코딩하는 단계(S22, 2 : t_embed)와, 매칭 정보와 데이터베이스를 인코딩하는 단계(S23, 3 : d_embed)와, 인코딩된 자연어 및 데이터베이스를 입력으로 받아 제1신경망을 통해 컬럼 및 테이블들의 후보들을 반환하는 단계(S24, 4 : cols_tabels_candidates)와, 자연어 문장과 컬럼 및 테이블 후보들을 입력받아 인코딩하는 단계(S25, 6 : candidate_embedding)와, 인코딩된 벡터를 받아 컬럼 및 테이블 후보가 얼마나 자연어 문장에 맞는지 나타내는 점수를 구하고, 반환하는 단계(S26, 7 : scor, 8 : return score)와, 모든 컬럼 및 테이블 후보들의 점수를 확인하여 가장 높은 점수의 컬럼 및 테이블 후보를 결정하는 단계(S27, 9 : best_candidate)를 포함한다.

[0043] 이하, 상기와 같이 구성되는 본 발명의 바람직한 실시예에 따른 신경망 기반 자연어로부터 SQL 질의 번역 시 사용되는 컬럼 및 테이블을 예측하는 방법에 대하여 좀 더 상세히 설명한다.

[0044] 먼저, S21 단계와 같이 자연어 문장(t)과 데이터베이스(d)를 입력받는다. 이때, 자연어 문장(t)은 텍스트 입력, 음성의 입력, 필기체 입력 등 다양한 방식의 입력일 수 있으며, 이미지 입력에서 텍스트를 추출한 결과일 수 있다.

[0045] 데이터베이스(d)의 입력은 본 발명을 수행하는 컴퓨터의 마이크로컴퓨터에서 적용 가능한 통신을 이용하여 데이터베이스에서 자연어 문장(t)을 이루는 단어들을 검색하고, 그 결과를 입력받는 것으로 한다.

[0046] 따라서 자연어 문장의 입력에 따라 자연어 문장과 데이터베이스를 단어 단위로 비교하여 매칭하게 된다.

[0047] 좀 더 구체적으로, 자연어 데이터베이스 매칭 과정에서는 m 개의 단어를 포함한 문장 S=(w1,w2,..., wm)과 컬럼들 C=(c1, c2,..., cn) 및 테이블들 T=(t1, t2, ... , tl)을 포함하는 데이터베이스가 주어졌을 때, 자연어 문장의 각 단어들과 컬럼, 테이블들의 매칭 정보 태그인 S_I=(w_i1, w_i2, ... , w_im), C_I=(c_i1, c_i2, ... , c_in), T_I=(t_i1, t_i2, ... , t_il)를 구한다.

[0048] 그 다음, 테이블 tx의 이름이 자연어에 (wa, wa+1,... ,wb)로서 등장하면, t x 에 해당하는 태그인 t_ix에 “[exact match]”를 할당한다. 또한 (wa, wa+1, ..., wb)에 해당하는 태그인 (w_ia, w_ia+1, ..., w_ib)에는 각각 “[table]”을 할당한다.

[0049] 만약 (wa, wa+1, ... , wb)가 tx의 이름의 부분문자열로서 등장하면 t_ix에 “[partial match]”를 할당한다.

[0050] 컬럼의 이름을 사용하여도 마찬가지로 “[exact match]”, “[partial match]”태그와 “[column]”태그를 할당한다.

[0051] 컬럼에 들어 있는 값과 자연어가 매칭되었을 때도 “[exact match]”, “[partial match]”와 “[cell]” 태그를 할당한다.

[0052] 태그를 할당하는 우선순위는 “[exact match]”가 “[partial match]”보다 높으며, “[table]”이 “[column]”보다 높으며, “[column]”이 “[cell]”보다 높다.

[0053] 정의되지 않은 태그에는 우선순위가 가장 낮은 “[none]”을 할당한다.

[0054] 도 3에서 1 : matching_info: = DATABASE_LINKING(t, d)는 자연어 문장(t)과 데이터베이스(d)를 매칭하는 함수이다.

[0055] 그 다음, S22단계와 같이 상기 S21단계의 매칭 정보와 자연어 문장(t)을 인코딩한다. 또한 S23단계와 같이 상기 S21단계의 매칭 정보와 데이터베이스(d)를 인코딩한다.

[0056] 도 3의 2 : t_embed : = SENTENCE_EMBDDING(t, matching_info)는 매칭 정보와 자연어 문장을 인코딩하는 함수이고, 도 3의 3 : d_embed := DATABASE_EMBEDDING(d , matching_info)는 매칭 정보와 데이터베이스를 인코딩하는 함수이다.

[0057] 자연어 및 데이터베이스 임베딩(또는 인코딩) 과정에서는 문장 S=(w1, w2, ..., wm), 컬럼들 C=(c1 , c2, ... ,cn) 및 테이블들 T=(t1, t2, ... , tl) 및 그 에 해당하는 매칭 정보 S_I=(w_i1, w_i2, ... , w_im), C_I=(c_i1, c_i2, ... ,c_in), T_I=(t_i1, t_i2, ..., t_il)를 이용하여 각 단어에 해당하는 임베딩 벡터들 S_E= (w_e1, w_e2, ... ,w_em), 각 컬럼에 해당하는 임베딩 벡터들 C_E=(c_e1, c_e2, ..., c_en) 및 각 테이블에 해당하는 임베딩 벡터들 T_E=(t_e1, t_e2, ... , t_el)를 구한다.

[0058] 우선 문장과 컬럼들의 이름 및 테이블들의 이름을 연결하여 사전 학습된 트랜스포머 신경망을 통해 각 단어, 컬럼의 단어, 테이블들의 단어에 대한 임베딩 벡터를 구한다.

[0059] 이후, 한 컬럼이나 한 테이블에 속하는 단어들끼리는 임베딩 벡터들끼리의 합 연산을 통해 컬럼 및 테이블에 해당하는 벡터를 구한다.

[0060] 각 단어, 컬럼 및 테이블에 해당하는 임베딩 벡터는 태그 정보의 임베딩 벡터와의 합 연산을 통해 최종 임베딩 벡터를 구한다.

[0061] 이처럼 임베딩 벡터를 구하는 과정을 도 4에 도시하였다.

[0062] 그 다음, S24단계와 같이 인코딩된 자연어 및 데이터베이스를 입력으로 받아 제1신경망을 통해 컬럼 및 테이블들의 후보들을 반환한다.

[0063] 도 3의 4 : cols_tables_candidates := CANDIDATE_GENERATE (t_embed, d_embed)는 인코딩된 자연어 및 데이터베이스를 입력으로 받아 제1신경망을 통해 컬럼 및 테이블들의 후보를 결정하는 함수이다.

[0064] 컬럼 및 테이블 후보들의 생성 과정은 제1신경망을 이용해 위에서 인코딩한 자연어, 컬럼 및 테이블들을 입력으로 받아 컬럼 및 테이블의 후보들을 생성한다.

[0065] 제1신경망은 자연어, 컬럼 및 테이블을 인코딩할 때 사용한 사전 학습된 트랜스포머 모델이 반환하는 첫 번째 벡터를 LSTM(Long Short-Term Memory)의 첫 번째 입력으로 사용한다.

[0066] 이 LSTM이 첫 번째로 반환하는 히든 스테이트(hidden state)는 세 개의 액션 임베딩 AC, AT, AE 와 내적 연산 및 소프트맥스된다.

[0067] 세 개의 액션은 각각 신경망이 다음으로 예측하는 것이 컬럼일지, 테이블일지, 혹은 예측을 끝낼 것인지를 의미하며, 소프트맥스된 점수가 가장 높은 액션이 선택된다.

[0068] AE가 선택된 경우 예측을 종료하며, AC, AT 가 선택된 경우 선택된 액션의 임베딩이 LSTM의 다음 입력으로 들어간다.

[0069] 다음으로 LSTM이 반환하는 히든 스테이트는 이전에 선택된 액션이 AC인 경우 컬럼들의 임베딩과 내적하여 소프트맥스되고, AT 인 경우 테이블의 임베딩과 내적하여 소프트맥스된다.

[0070] 소프트맥스 점수가 가장 높은 컬럼 및 테이블이 선택되며, 해당 컬럼 및 테이블의 임베딩은 LSTM의 입력으로 들어가 LSTM은 다시 액션을 선택하는 과정을 반복하면서 컬럼 및 테이블들을 고르게 된다.

[0071] LSTM이 순차적으로 컬럼과 테이블을 고르는 과정에 빔 서치를 적용하여 컬럼 및 테이블의 후보들을 선택한다.

[0072] 그 다음, S25단계에서는 자연어 문장과 컬럼 및 테이블 후보들을 입력받아 인코딩한다.

[0073] 도 3의 5: def GET_CANDIDATE_SCORE(candidate); 6: candidate_embedding := CANDIDATE_EMBEDDING (t, candidate)는 자연어 문장과 컬럼 및 테이블 후보를 인코딩하는 함수이다.

[0074] 좀 더 구체적으로 S25단계에서는 자연어 및 컬럼 및 테이블의 후보 하나를 입력으로 받아 정답일 확률을 반환하는 신경망을 통해 상기 과정에서 생성된 각 후보들을 랭킹한다.

[0075] 이 과정에서는 스키마를 테이블 및 컬럼이 정점이며, 그것들의 관계를 간선으로 하는 그래프로 여긴다.

[0076] 여기서 관계는 각 테이블이 어떤 컬럼들을 포함하는지와 그 반대의 관계, 컬럼이 외래키인 경우 어떤 컬럼의 외래키인지와 그 반대의 관계이다.

[0077] 상기 과정에서 생성한 각 후보는 스키마 그래프 상에서 연결된 그래프가 되도록 스타이너 트리(steiner tree)알고리즘을 이용해 후보들에 컬럼 및 테이블들을 추가한다.

[0078] 그 다음, S26단계에서는 인코딩된 벡터를 받아 컬럼 및 테이블 후보가 얼마나 자연어 문장에 맞는지 나타내는 점수를 구하고 반환한다.

[0079] 도 3의 7: score := CANDIDATE_CHECK (candidate_embedding), 8: return score는 인코딩된 벡터를 받아 해당 후보가 얼마나 자연어 문장에 맞는지를 나타내는 점수를 반환하는 함수와 스코어를 반환하는 명령이다.

[0080] 좀 더 구체적으로, 자연어와 후보 하나에 들어있는 컬럼 및 테이블들의 이름들을 이어 붙여 사전 학습된 제2신경망인 트랜스포머 신경망에 입력으로 넣어 준다.

[0081] 트랜스포머 신경망이 반환하는 첫 번째 벡터에 다중 퍼셉트론 신경망을 적용하여 시그모이드(sigmoid) 함수를 적용해 0부터 1 사이의 실수값을 반환하게 한다.

[0082] 상기 과정을 수행하는 신경망의 학습 방법은 학습 데이터로부터 참인 후보와 거짓인 후보를 추출하여 이루어진다. 참인 후보는 정답 SQL에 사용되는 테이블 및 컬럼들로 만든다. 참인 후보는 신경망이 반환하는 실수와 값 1의 이진 크로스 엔트로피(binary cross entropy) 함수를 로스 함수로서 학습한다.

[0083] 거짓인 후보는 스키마 그래프로부터 연결된 부분 그래프들 중 참인 그래프가 아닌 그래프를 무작위로 추출하여 사용한다. 거짓인 후보는 신경망이 반환하는 실수와 값 0의 이진 크로스 엔트로피 함수를 로스 함수로서 학습한다.

[0084] 학습된 참 후보와 거짓 후보의 비율이 같도록 반복적으로 학습하여 신경망을 학습한다.

[0085] 그 다음, S27단계에서는 모든 컬럼 및 테이블 후보들의 점수를 확인하여 가장 높은 점수의 컬럼 및 테이블 후보를 결정한다.

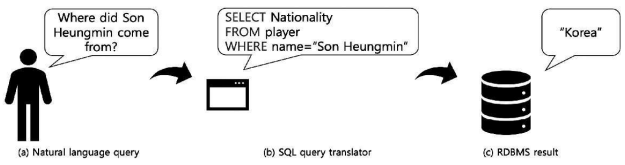
[0086] 도 3의 9 : best_candidate := max(cols_tables_candidates, key=GET_CANDIDATE_SCORE)는 앞서 S26단계에서 각 컬럼 및 테이블의 자연어 문장과의 일치도가 가장 높은 컬럼 및 테이블을 선택하는 함수이다.

[0087] 이와 같은 과정을 통해 후보들 중 가장 일치도가 높은 컬럼 및 테이블을 선택할 수 있어, 정확도를 높일 수 있게 된다.

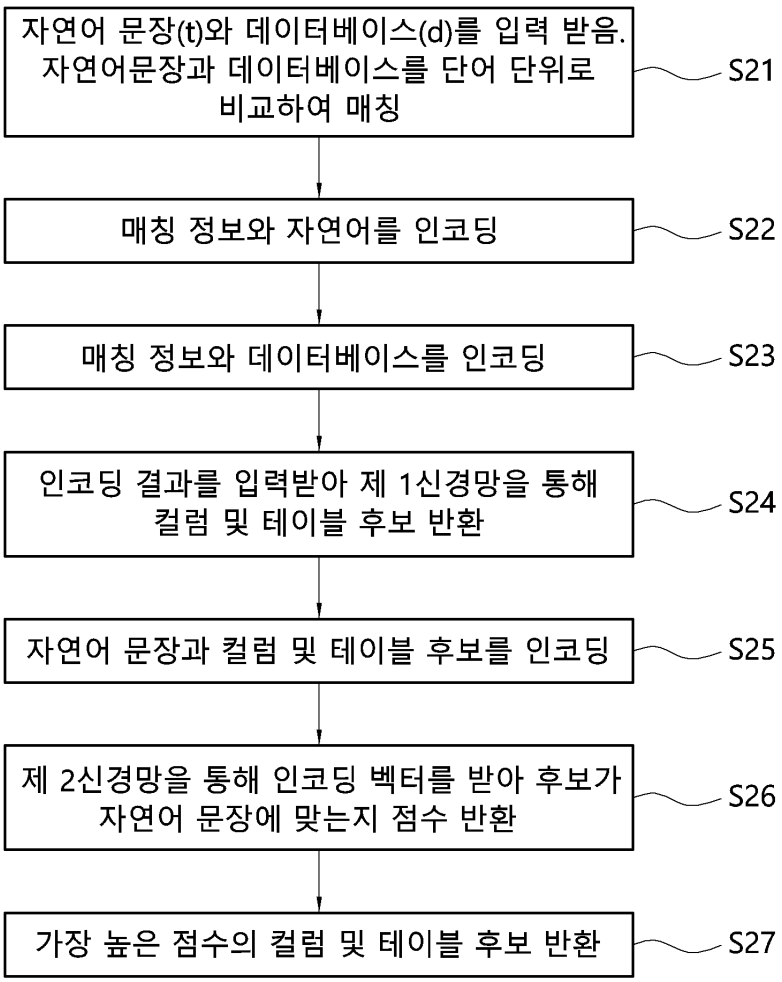
[0088] 본 발명은 상기 실시예에 한정되지 않고 본 발명의 기술적 요지를 벗어나지 아니하는 범위 내에서 다양하게 수정, 변형되어 실시될 수 있음은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 있어서 자명한 것이다.

도면

도면1



도면2



도면3

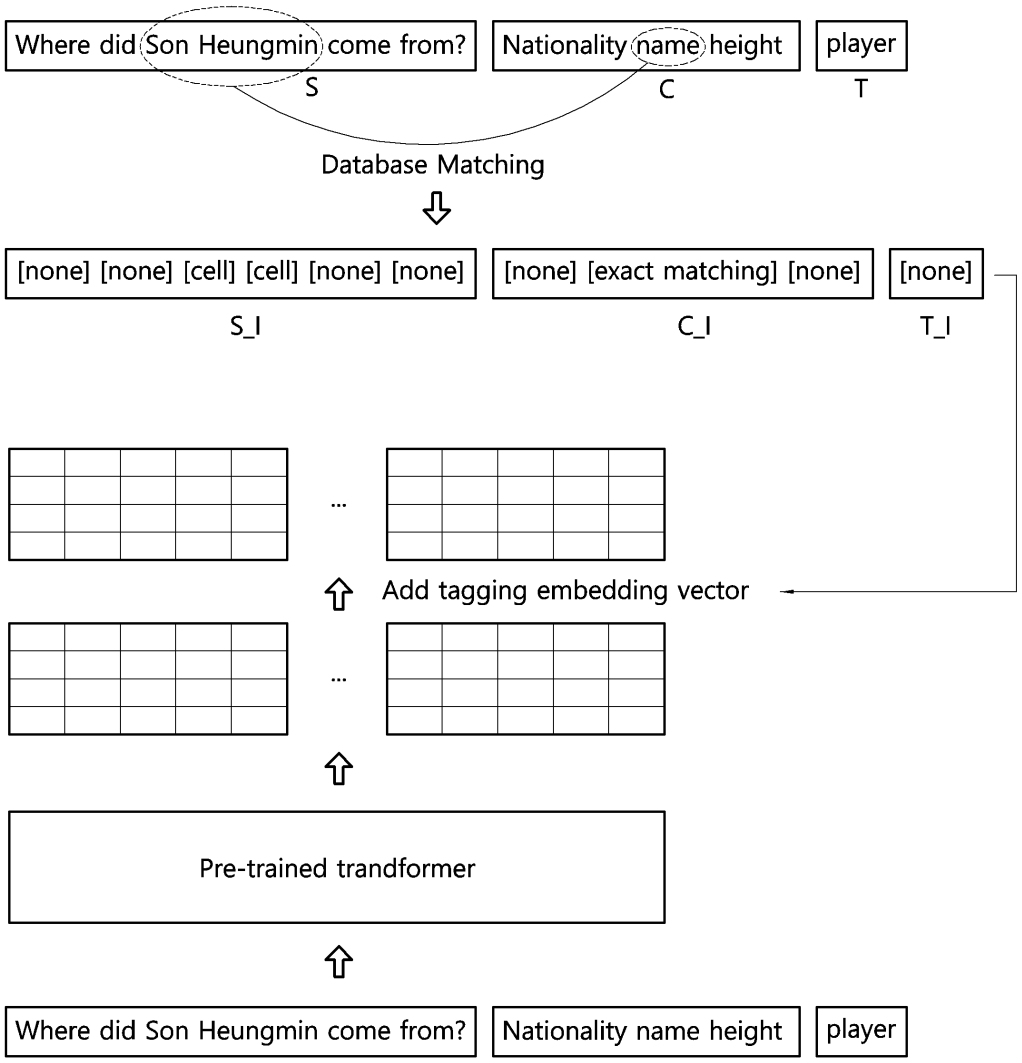
Algorithm 1 ColumnTablePrediction(*t*, *d*)

Input: a text string *t*, a database *d*

Output: columns and tables best_candidate

```
matching_info := DATABASE_LINKING(t, d);
t_embed := SENTENCE_EMBEDDING(t, matching_info)
d_embed := DATABASE_EMBEDDING(d, matching_info)
cols_tables_candidates := CANDIDATE_GENERATE(t_embed, d_embed)
def GET_CANDIDATE_SCORE(candidate):
    candidate_embedding := CANDIDATE_EMBEDDING(t, candidate)
    score := CANDIDATE_CHECK(candidate_embedding)
    return score
best_candidate := max(cols_tables_candidates, key=GET_CANDIDATE_SCORE)
```

도면4


Where did Son Heungmin come from?
Nationality name height
player
S
C
T
Database Matching
[none] [none] [cell] [cell] [none] [none]
[none] [exact matching] [none]
[none]
S_I
C_I
T_I
...
Add tagging embedding vector
...
Pre-trained trandformer
Where did Son Heungmin come from?
Nationality name height
player