



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0083415
(43) 공개일자 2022년06월20일

- | | |
|--|--|
| <p>(51) 국제특허분류(Int. Cl.)
 <i>G06F 16/2455</i> (2019.01) <i>G06F 16/242</i> (2019.01)
 <i>G06F 16/2458</i> (2019.01) <i>G06F 16/28</i> (2019.01)
 <i>G06F 16/36</i> (2019.01)</p> <p>(52) CPC특허분류
 <i>G06F 16/2456</i> (2019.01)
 <i>G06F 16/2433</i> (2019.01)</p> <p>(21) 출원번호 10-2020-0173639
 (22) 출원일자 2020년12월11일
 심사청구일자 2020년12월11일</p> | <p>(71) 출원인
 경희대학교 산학협력단
 경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)</p> <p>(72) 발명자
 이영구
 경기도 수원시 영통구 청명북로 33, 431동 1704호(영통동, 청명마을삼성아파트)</p> <p>우메르 쿠두스
 경기도 수원시 영통구 청명북로15번길 13, B01호(영통동)</p> <p>(74) 대리인
 특허법인도담</p> |
|--|--|

전체 청구항 수 : 총 4 항

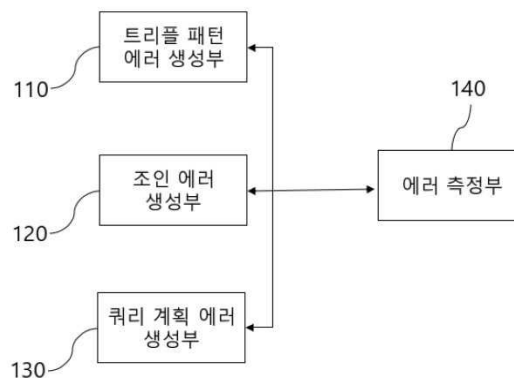
(54) 발명의 명칭 SPARQL 쿼리 처리 엔진 평가 방법 및 이를 실행하는 장치

(57) 요약

본 발명의 일 실시예에 따른 SPARQL 쿼리 처리 엔진 평가 장치에서 실행되는 SPARQL 쿼리 처리 엔진 평가 방법은 트리플 패턴 각각의 실제 카디널리티 벡터 및 상기 트리플 패턴 각각의 추정 카디널리티 벡터를 이용하여 전체 트리플 패턴 에러를 생성하는 단계, 조인의 실제 카디널리티 벡터 및 상기 조인의 추정 카디널리티 벡터를 이용하여 조인 에러를 생성하는 단계, 상기 트리플 패턴의 셋에 대한 실제 카디널리티 벡터 및 상기 트리플 패턴의 셋에 대한 추정 카디널리티 벡터를 이용하여 쿼리 계획 에러를 생성하는 단계 및 상기 전체 트리플 패턴 에러, 상기 조인 에러 및 상기 쿼리 계획 에러를 이용하여 상기 트리플 패턴의 카디널리티 추정 결과에 대한 에러를 측정하는 단계를 포함한다.

대표도 - 도1

100



(52) CPC특허분류

G06F 16/2458 (2019.01)

G06F 16/284 (2019.01)

G06F 16/36 (2019.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711111586
과제번호	2018R1A2A2A05023669
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구(과기정통부)(R&D)
연구과제명	대용량 RDF그래프 요약 및 압축 기반 고속 RDF질의 처리 기술
기 여 율	1/1
과제수행기관명	경희대학교(국제캠퍼스)
연구기간	2020.03.01 ~ 2021.02.28

명세서

청구범위

청구항 1

SPARQL 쿼리 처리 엔진 평가 장치에서 실행되는 SPARQL 쿼리 처리 엔진 평가 방법에 있어서,

트리플 패턴 각각의 실제 카디널리티 벡터 및 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴 각각의 추정 카디널리티 벡터를 이용하여 전체 트리플 패턴 에러를 생성하는 단계;

상기 트리플 패턴 중 두 개의 패턴에 대한 조인의 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 조인의 추정 카디널리티 벡터를 이용하여 조인 에러를 생성하는 단계;

상기 트리플 패턴의 셋에 대한 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴의 셋에 대한 추정 카디널리티 벡터를 이용하여 쿼리 계획 에러를 생성하는 단계; 및

상기 전체 트리플 패턴 에러, 상기 조인 에러 및 상기 쿼리 계획 에러를 이용하여 상기 트리플 패턴의 카디널리티 추정에서 전체 오류를 측정하고, 트리플 패턴과 전체 쿼리 계획 에러를 결합하여 상기 SPARQL 쿼리 처리 엔진을 평가하는 단계를 포함하는 것을 특징으로 하는

SPARQL 쿼리 처리 엔진 평가 방법.

청구항 2

제1항에 있어서,

상기 트리플 패턴의 카디널리티 추정에서 전체 오류를 측정하고, 트리플 패턴과 전체 쿼리 계획 에러를 결합하여 평가하는 단계는

쿼리 계획에 사용된 상기 조인에서 발생한 전체 쿼리 계획 에러 및 상기 트리플 패턴의 상기 추정 카디널리티 벡터를 생성하는 과정에서 상기 각각의 트리플 패턴을 결합하는 단계를 더 포함하는 것을 특징으로 하는

SPARQL 쿼리 처리 엔진 평가 방법.

청구항 3

SPARQL 쿼리 처리 엔진 평가 장치에 있어서,

트리플 패턴 각각의 실제 카디널리티 벡터 및 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴 각각의 추정 카디널리티 벡터를 이용하여 전체 트리플 패턴 에러를 생성하는 트리플 패턴 에러 생성부;

상기 트리플 패턴 중 두 개의 패턴에 대한 조인의 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 조인의 추정 카디널리티 벡터를 이용하여 조인 에러를 생성하는 조인 에러 생성부;

상기 트리플 패턴의 셋에 대한 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴의 셋에 대한 추정 카디널리티 벡터를 이용하여 쿼리 계획 에러를 생성하는 쿼리 계획 에러 생성부;

상기 전체 트리플 패턴 에러, 상기 조인 에러 및 상기 쿼리 계획 에러를 이용하여 상기 트리플 패턴의 카디널리티 추정에서 전체 오류를 측정하고, 트리플 패턴과 전체 쿼리 계획 에러를 결합하여 상기 SPARQL 쿼리 처리 엔진을 평가하는 에러 측정부를 포함하는 것을 특징으로 하는

SPARQL 쿼리 처리 엔진 평가 장치.

청구항 4

제3항에 있어서,

상기 에러 측정부는

쿼리 계획에 사용된 상기 조인에서 발생한 전체 쿼리 계획 에러 및 상기 트리플 패턴의 상기 추정 카디널리티 벡터를 생성하는 과정에서 상기 각각의 트리플 패턴을 결합하는 것을 특징으로 하는

SPARQL 쿼리 처리 엔진 평가 장치.

발명의 설명

기술 분야

- [0001] 본 발명은 SPARQL 쿼리 처리 엔진 평가 방법 및 이를 실행하는 장치에 관한 것으로, 보다 상세하게는 카디널리티 추정기의 성능을 효율적으로 평가할 수 있는 SPARQL 쿼리 처리 엔진 평가 방법 및 이를 실행하는 장치에 관한 것이다.

배경 기술

- [0003] 시맨틱 웹(Semantic Web)이란 현재 사용되는 인터넷 등과 같은 분산 환경에서 리소스, 예컨대 웹문서, 각종 파일, 각종 응용 서비스 등에 관한 정보와 자원 사이의 관계-의미 정보(semantics)를 기계적으로 처리할 수 있는 온톨로지(ontology) 정보로서 표현하는 기술을 의미한다.
- [0004] 관계형 데이터 베이스는 데이터 관리를 위하여 가장 보편적으로 사용되고 있는 모델이며, 구조적 질의어(SQL: Structured Query Language)는 관계형 데이터 베이스의 데이터에 신호를 보내 문의하고(interrogate) 처리하기 위해 사용되는 언어를 의미한다.
- [0005] 따라서, 메인 프레임들과 관련하여 사용하기 위해 개발된 클라이언트/서버 환경들을 위해 설계된 대부분의 데이터베이스 관리 시스템들(DBMSs: Database Management Systems)은 구조적 질의어를 지원하고 있다.
- [0006] 스파클(SPARQL)은 시맨틱 웹에서 사실상 표준으로 자리잡은 질의어이며, 시맨틱 웹 서비스의 저장소로써 관계형 데이터베이스를 사용하는 경우 SPARQL-TO-SQL 변환기가 반드시 필요하다. SPARQL-TO-SQL 변환기는 내부적으로 SPARQL 구문분석기와 변환 알고리즘으로 구성된다.
- [0007] 일반적으로, 시맨틱 웹 서비스에 관계형 데이터베이스를 사용하는 이유는 안정성과 대용량 처리에 따른 어려움을 해소하기 위해서이다. 따라서, SPARQL 언어로 표현된 질의문은 SPARQL-TO-SQL 변환기를 거쳐 생성된 SQL문을 통해 데이터베이스에서 필요한 정보를 불러오게 된다.
- [0008] 스파클 질의문과 SQL문은 자료를 기술하는 관점이나 언어적 특성이 다르기 때문에 변환 과정을 통해 본래 의도했던 것과 다른 결과물이 나오는 에러를 범할 가능성이 있다.

발명의 내용

해결하려는 과제

- [0010] 본 발명은 카디널리티 추정기의 성능을 효율적으로 평가할 수 있는 비용 기반 페더레이션 SPARQL 쿼리 엔진 벤치마킹을 실행할 수 있도록 하는 SPARQL 쿼리 처리 엔진 평가 방법 및 이를 실행하는 장치를 제공하는 것을 목적으로 한다.
- [0011] 또한, 본 발명은 페더레이션 엔진의 성능을 평가하기 위해 전체 쿼리 런타임에서 카디널리티 에러를 추정할 수 있는 SPARQL 쿼리 처리 엔진 평가 방법 및 이를 실행하는 장치를 제공하는 것을 목적으로 한다.
- [0013] 본 발명의 목적들은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 본 발명의 다른 목적 및 장점들은 하기의 설명에 의해서 이해될 수 있고, 본 발명의 실시예에 의해 보다 분명하게 이해될 것이다. 또한, 본 발

명의 목적 및 장점들은 특허 청구 범위에 나타낸 수단 및 그 조합에 의해 실현될 수 있음을 쉽게 알 수 있을 것이다.

과제의 해결 수단

[0015] 이러한 목적을 달성하기 위한 SPARQL 쿼리 처리 엔진 평가 장치에서 실행되는 SPARQL 쿼리 처리 엔진 평가 방법은 트리플 패턴 각각의 실제 카디널리티 벡터 및 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴 각각의 추정 카디널리티 벡터를 이용하여 전체 트리플 패턴 에러를 생성하는 단계, 상기 트리플 패턴 중 두 개의 패턴에 대한 조인의 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 조인의 추정 카디널리티 벡터를 이용하여 조인 에러를 생성하는 단계, 상기 트리플 패턴의 셋에 대한 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴의 셋에 대한 추정 카디널리티 벡터를 이용하여 쿼리 계획 에러를 생성하는 단계 및 상기 전체 트리플 패턴 에러, 상기 조인 에러 및 상기 쿼리 계획 에러를 이용하여 상기 트리플 패턴의 카디널리티 추정에서 전체 오류를 측정하고, 트리플 패턴과 전체 쿼리 계획 에러를 결합하여 상기 SPARQL 쿼리 처리 엔진을 평가하는 단계를 포함한다.

발명의 효과

[0017] 전술한 바와 같은 본 발명에 의하면, 카디널리티 추정기의 성능을 효율적으로 평가할 수 있는 비용 기반 페더레이션 SPARQL 쿼리 엔진 벤치마킹을 실행할 수 있다는 장점이 있다.

[0018] 또한 본 발명에 의하면, 페더레이션 엔진의 성능을 평가하기 위해 전체 쿼리 런타임에서 카디널리티 에러를 추정할 수 있다는 장점이 있다.

도면의 간단한 설명

[0020] 도 1은 본 발명의 일 실시예에 따른 SPARQL 쿼리 처리 엔진 평가 장치를 설명하기 위한 블록도이다.

도 2는 본 발명에 따른 SPARQL 쿼리 처리 엔진 평가 방법의 일 실시예를 설명하기 위한 흐름도이다.

도 3은 본 발명에 따른 SPARQL 쿼리 처리 엔진 평가 과정을 설명하기 위한 예시도이다.

발명을 실시하기 위한 구체적인 내용

[0021] 전술한 목적, 특징 및 장점은 첨부된 도면을 참조하여 상세하게 후술되며, 이에 따라 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 상세한 설명을 생략한다. 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시예를 상세히 설명하기로 한다. 도면에서 동일한 참조부호는 동일 또는 유사한 구성요소를 가리키는 것으로 사용된다.

[0023] 도 1은 본 발명의 일 실시예에 따른 SPARQL 쿼리 처리 엔진 평가 장치를 설명하기 위한 블록도이다.

[0024] 도 1을 참조하면, SPARQL 쿼리 처리 엔진 평가 장치(100)는 트리플 패턴 에러 생성부(110), 조인 에러 생성부(120), 쿼리 계획 에러 생성부(130) 및 에러 측정부(140)를 포함한다.

[0025] 트리플 패턴 에러 생성부(110)는 트리플 패턴 각각의 실제 카디널리티 벡터 및 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴 각각의 추정 카디널리티 벡터를 이용하여 전체 트리플 패턴 에러(ET)를 생성한다.

[0026] 전체 트리플 패턴 에러(ET)를 정의하기 위해 Q는 트리플 패턴(TP1, TP2 및 TP3)을 포함하는 SPARQL 쿼리를 의미하고, $\text{vec}(r) = (\text{Cr}(TP1), \dots, \text{Cr}(TPn)) \in R$: T의 실제 카디널리티 벡터를 의미하고, $\text{vec}(e) = (\text{Ce}(TP1), \dots, \text{Ce}(TPn)) \in R$: T의 추정 카디널리티 벡터라고 하면, 전체 트리플 패턴 에러(ET)은 [수학식 1]와 같이 정의된다.

[0028] [수학식 1]

$$E_T = 2 * \frac{|\vec{r}-\vec{e}|}{|\vec{r}|+|\vec{e}|}$$

$$= 2 * \frac{\sqrt{\sum_{i=1}^n (Cr(TP_i)-Ce(TP_i))^2}}{\sqrt{\sum_{i=1}^n (Cr(TP_i))^2}+\sqrt{\sum_{i=1}^n (Ce(TP_i))^2}}$$

[0029]

[0031] 조인 에러 생성부(120)는 트리플 패턴 중 두 개의 패턴에 대한 조인의 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 조인의 추정 카디널리티 벡터를 이용하여 조인 에러를 생성한다.

[0032] 조인 에러(EJ)를 정의하기 위해 Q는 상기 트리플 패턴 중 두 개의 패턴에 대한 조인 $J = \{J_1, \dots, J_n\}$ 을 포함하는 SPARQL 쿼리를 의미하고, $\text{vec}(r) = (Cr(J_1), \dots, Cr(J_n)) \in R$: 조인(J)의 실제 카디널리티 벡터를 의미하고, $\text{vec}(e) = (Ce(J_1), \dots, Ce(J_n)) \in J$: 조인(J)의 추정 카디널리티 벡터를 의미한다고 하면, 조인 에러(EJ)는 [수학식 2]과 같이 정의된다.

[0034] [수학식 2]

$$E_J = 2 * \frac{|\vec{r}-\vec{e}|}{|\vec{r}|+|\vec{e}|}$$

$$= 2 * \frac{\sqrt{\sum_{i=1}^n (Cr(J_i)-Ce(J_i))^2}}{\sqrt{\sum_{i=1}^n (Cr(J_i))^2}+\sqrt{\sum_{i=1}^n (Ce(J_i))^2}}$$

[0035]

[0037] 쿼리 계획 에러 생성부(130) 트리플 패턴의 셋에 대한 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴의 셋에 대한 추정 카디널리티 벡터를 이용하여 쿼리 계획 에러를 생성한다.

[0038] 쿼리 계획 에러(EP)를 정의하기 위해 Q는 SPARQL 쿼리를 의미하고, $\text{vec}(r) = (r_1, \dots, r_n) \in R$: 트리플 패턴의 셋(TJ)의 실제 카디널리티 벡터, $\text{vec}(e) = (e_1, \dots, e_n) \in R$: 트리플 패턴의 셋(TJ)의 추정 카디널리티 벡터를 의미한다고 하면, 쿼리 계획 에러(EP)는 [수학식 3]과 같이 정의된다.

[0040] [수학식 3]

$$E_P = 2 * \frac{|\vec{r}-\vec{e}|}{|\vec{r}|+|\vec{e}|}$$

$$= 2 * \frac{\sqrt{\sum_{i=1}^n (Cr(r_i)-Ce(e_i))^2}}{\sqrt{\sum_{i=1}^n (Cr(r_i))^2}+\sqrt{\sum_{i=1}^n (Ce(e_i))^2}}$$

[0041]

[0043] 에러 측정부(140)는 전체 트리플 패턴 에러(ET), 상기 조인 에러(EJ) 및 상기 쿼리 계획 에러(EP)를 이용하여 상기 트리플 패턴(TP1, TP2 및 TP3)의 카디널리티 추정에서 전체 오류를 측정하고, 트리플 패턴(TP1, TP2 및 TP3)과 전체 쿼리 계획 에러를 결합하여 상기 SPARQL 쿼리 처리 엔진을 평가한다.

[0044] 이때, 에러 측정부(140)는 쿼리 계획에 사용된 상기 조인에서 발생된 전체 쿼리 계획 에러 및 상기 트리플 패턴의 상기 추정 카디널리티 벡터를 생성하는 과정에서 상기 각각의 트리플 패턴을 결합하여 SPARQL 쿼리 처리 엔진을 평가할 수 있다.

[0046] 도 2는 본 발명에 따른 SPARQL 쿼리 처리 엔진 평가 방법의 일 실시예를 설명하기 위한 흐름도이다.

[0047] 도 2를 참조하면, SPARQL 쿼리 처리 엔진 평가 장치(100)는 트리플 패턴 각각의 실제 카디널리티 벡터 및 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴 각각의 추정 카디널리티 벡터를 이용하여 전체 트리플

패턴 에러를 생성한다(단계 S210).

- [0048] SPARQL 쿼리 처리 엔진 평가 장치(100)는 상기 트리플 패턴 중 두 개의 패턴에 대한 조인의 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 조인의 추정 카디널리티 벡터를 이용하여 조인 에러를 생성한다(단계 S220).
- [0049] SPARQL 쿼리 처리 엔진 평가 장치(100)는 상기 트리플 패턴의 셋에 대한 실제 카디널리티 벡터 및 상기 SPARQL 쿼리 처리 엔진에 의해 추정된 상기 트리플 패턴의 셋에 대한 추정 카디널리티 벡터를 이용하여 쿼리 계획 에러를 생성한다(단계 S230).
- [0050] SPARQL 쿼리 처리 엔진 평가 장치(100)는 상기 전체 트리플 패턴 에러, 상기 조인 에러 및 상기 쿼리 계획 에러를 이용하여 상기 트리플 패턴의 카디널리티 추정에서 전체 오류를 측정하고, 트리플 패턴과 전체 쿼리 계획 에러를 결합하여 상기 SPARQL 쿼리 처리 엔진을 평가한다(단계 S240).
- [0051] 단계 S240에 대한 일 실시예에서, SPARQL 쿼리 처리 엔진 평가 장치(100)는 쿼리 계획에 사용된 상기 조인에서 발생한 전체 쿼리 계획 에러 및 상기 트리플 패턴의 상기 추정 카디널리티 벡터를 생성하는 과정에서 상기 각각의 트리플 패턴을 결합하여 SPARQL 쿼리 처리 엔진을 평가할 수 있다.
- [0053] 도 3은 본 발명에 따른 SPARQL 쿼리 처리 엔진 평가 과정을 설명하기 위한 예시도이다.
- [0054] 도 3을 참조하면, 트리플 패턴(TP1, TP2 및 TP3)과 두 개의 조인이 있는 SPARQL 쿼리를 포함한다. 참조번호 3(a)는 SPARQL 쿼리의 트리플 패턴 중 패턴에 대한 실제 카디널리티(Cr) 및 추정 카디널리티(참조번호 3(b)와 같이 엔진 1의 경우 Ce1 및 참조번호 3(c)와 같이 엔진 2의 경우 Ce2)와 를 나타낸다.
- [0055] 트리플 패턴(TP1, TP2 및 TP3)의 질의를 위해 두 엔진 모두 가장 작은 기수를 가진 트리플 패턴(TP1, TP2 및 TP3)을 선택하여 왼쪽 깊이의 쿼리 계획을 생성한다고 가정해보자. 이 결합의 결과는 남은 세 번째 트리플 패턴으로 두 번째 결합을 수행하는 데 사용된다.
- [0056] 실제 기본값을 사용하여 최적의 쿼리 실행 계획은 먼저 트리플 패턴(TP1, TP2 및 TP3) 중 두 개의 패턴(TP1과 TP2) 사이의 조인을 수행한 다음 나머지 하나의 패턴(TP3)과 두 번째 조인을 수행하는 것이다.
- [0057] 참조번호 도 3(b)에서 보여진 것과 같이 엔진 1에 의해 동일한 계획인 생성된다. 엔진 2에 의해 생성된 계획은 참조번호 도 3(c)에 제시되어 있다.
- [0058] 트리플 패턴(TP1, TP2 및 TP3)과 트리플 패턴(TP1, TP2 및 TP3) 사이의 조인에 대한 카디널리티 추정에서 큰 에러가 발생하여 엔진 2가 최적의 계획을 선택하지 않았다는 것이다.
- [0059] 동기 부여의 예는 좋은 카디널리티 추정치가 더 나은 질의 계획을 생산하기 위해 필수적이라는 것을 분명히 보여준다.
- [0060] 이때, 카디널리티 추정의 정확성이 전체 쿼리 계획과 전체 쿼리 런타임 성능에 얼마나 영향을 미치는가에 대한 중요하다. 따라서, 본 발명은 전체 트리플 패턴 에러(ET), 조인 에러(EJ) 및 쿼리 계획 에러(EP)를 측정하기 위한 방법을 제안한다.
- [0061] 즉, 본 발명은 트리플 패턴(TP1, TP2 및 TP3)의 카디널리티 추정의 전반적인 에러를 측정하기 위해 Q-에러, 전체 트리플 패턴 에러(ET), 조인 에러(EJ) 및 쿼리 계획 에러(EP)를 정의한다.
- [0062] 이때, 트리플 패턴(TP1, TP2 및 TP3)의 카디널리티 추정에서 각각의 트리플 패턴(TP1, TP2 및 TP3) 및 전체 쿼리 계획 에러를 결합한다.
- [0063] 상기의 Q-에러는 실제 카디널리티 값으로부터 추정 카디널리티 값을 추정하기 위한 팩터이다. 이러한 Q-에러는 아래의 [수학식 4]과 같다.

[0065] [수학식 4]

Definition 1 (q-error). Let $\vec{r} = (r_1, \dots, r_n) \in \mathbb{R}$ where $r_i > 0$ be a vector of real values and $\vec{e} = (e_1, \dots, e_n) \in \mathbb{R}$ be the vector of the corresponding estimated values. By defining $\vec{e}/\vec{r} = \frac{\vec{e}}{\vec{r}} = (e_1/r_1, \dots, e_n/r_n)$, then q-error of estimation e of r is given as

$$\|e/r\|_Q = \max_{1 \leq i \leq n} \|e_i/r_i\|_Q, \text{ where}$$

$$\|e_i/r_i\|_Q = \max(e_i/r_i, r_i/e_i)$$

[0066]

[0068] 도 3에서와 같이 트리플 패턴(TP1, TP2 및 TP3) 중 제1 패턴(TP1)의 실제 카디널리티는 100(즉, Cr(TP1)=100)인 반면, 동일한 트리플 패턴에 대한 엔진 1의 추정 카디널리티는 90(즉, Cr(TP1)=90)이다. 따라서, 이 개별 트리플 패턴에 대한 q-에러는 최대(90/100, 100/90) = 1.11이다.

[0069] 도 3(b)와 같이 쿼리의 트리플 패턴(TP1, TP2 및 TP3)에 대한 전체 q-에러는 트리플 패턴(TP1, TP2 및 TP3)의 모든 q-에러 값의 최대값(1.11, 1.25, 1) = 1.25이다.

[0070] 전체 쿼리 계획의 q-에러는 트리플 패턴(TP1, TP2 및 TP3) 및 쿼리 계획에 사용되는 조인에서 최대 q-에러 값이 될 것이다. 즉, max(1.11, 1.25, 1, 1.3, 3) = 3이다.

[0071] q-에러는 절대적 또는 2차적 차이 대신 비율을 사용하므로 계획 결정을 내릴 때 상대적 차이만 중요하다는 직관을 포착할 수 있다. 또한, q-에러는 쿼리의 q-에러가 경계된 경우 계획 품질에 대한 이론적 상한을 제공한다. 계산된 값 중에서 최대값만 고려하기 때문에 이번 대책으로 평균 추정치가 좋은 계획은 부실하다고 간주될 수 있다.

[0072] 도 3(b) 및 도 3(c)에 제시된 쿼리 계획을 고려한다. 도 3(b) 및 도 3(c)은 트리플 패턴(TP1, TP2 및 TP3)의 q-에러이지만, 도 3(b)의 쿼리 계획이 최적인 반면, 도 3(c)의 쿼리 계획은 그렇지 않다.

[0073] 이러한 문제를 해결하기 위해, 본 발명은 아래에 정의된 추가 측정 기준을 제안한다.

[0074] 전체 트리플 패턴 에러(ET)를 정의하기 위해 Q는 트리플 패턴(TP1, TP2 및 TP3)을 포함하는 SPARQL 쿼리를 의미하고, $\text{vec}(r) = (\text{Cr}(TP1), \dots, \text{Cr}(TPn)) \in \mathbb{R}$: T의 실제 카디널리티 벡터를 의미하고, $\text{vec}(e) = (\text{Ce}(TP1), \dots, \text{Ce}(TPn)) \in \mathbb{R}$: T의 추정 카디널리티 벡터라고 하면, 전체 트리플 패턴 에러(ET)는 [수학식 2]와 같이 정의된다.

[0075] 도 1에 제시된 동기 부여 예에서 실제 카디널리티 벡터 = (100,200,300)와 엔진 1 추정 카디널리티 벡터 = (90,250,300)는 다음과 같다. 따라서 전체 트리플 패턴 에러(ET) = 2 20.0658 = 0.1316. 마찬가지로 엔진 2 추정 카디널리티 벡터는 = 200,500,600)이다. 따라서 엔진 2는 전체 트리플 패턴 에러(ET) = 2 ? 0.388 = 0.7765를 달성한다.

[0076] 조인 에러(EJ)를 정의하기 위해 Q는 조인 J = {J1, ..., Jn}을 포함하는 SPARQL 쿼리를 의미하고, $\text{vec}(r) = (\text{Cr}(J1), \dots, \text{Cr}(Jn)) \in \mathbb{R}$: 조인(J)의 실제 카디널리티 벡터를 의미하고, $\text{vec}(e) = (\text{Ce}(J1), \dots, \text{Ce}(Jn)) \in \mathbb{R}$: 조인(J)의 추정 카디널리티 벡터를 의미한다고 하면, 조인 에러(EJ)는 [수학식 3]과 같이 정의된다.

[0077] 쿼리 계획 에러(EP)를 정의하기 위해 Q는 SPARQL 쿼리를 의미하고, $\text{vec}(r) = (r1, \dots, rn) \in \mathbb{R}$: 트리플 패턴의 셋(TJ)의 실제 카디널리티 벡터, $\text{vec}(e) = (e1, \dots, en) \in \mathbb{R}$: 트리플 패턴의 셋(TJ)의 추정 카디널리티 벡터를 의미한다고 하면, 쿼리 계획 에러(EP)는 [수학식 4]과 같이 정의된다.

[0078] 도 3(b)에 제시된 것은 트리플 패턴(TP1, TP2 및 TP3)과 조인의 실제 카디널리티 벡터를 나타낸다. 즉, 도 3(b)에 제시된 것은 $\vec{r} = (100, 200, 300, 50, 50)$ 및 엔진 1에 의해 추정된 카디널리티 벡터 $\vec{e} = (90, 250, 300, 65, 150)$ 를 나타낸다.

[0079] 따라서, 엔진 1의 경우 쿼리 계획 에러(EP) = 2*0.1391 = 0.2784이고, 엔진 2가 쿼리 계획 에러(EP) = 2*0.3838 = 0.7676을 달성함 이러한 행렬에서 과소평가와 과소평가도 대칭적으로 처리 된다.

[0080] 본 발명에 따른 방법과 런타임 간의 상관 관계를 더 자세히 조사하기 위해, 쿼리 런타임과 처음 6개의 메트릭 각각에 해당하는 에러 사이의 Spearman의 상관 계수를 측정했다. 그 결과는 [표 1]에 나와 있다.

[0081] LargeRDFBench는 쿼리 데이터 셋을 사용하고 다른 쿼리 처리 엔진에 본 발명에 따른 방법을 적용하여 실험하였다. LargeRDFBench는 다양한 크기의 총 13 개의 실제 RDF 데이터 세트로 구성된다.

[0082] LargeRDFBench는 총 40 개의 쿼리로 구성되고, 40개의 쿼리는 FedBench의 14 개의 간단한 쿼리 (S1-S14), 10 개의 복잡한 쿼리 (C1-C10), 8 개의 복잡한 + 높은 소스 쿼리 (CH1-CH8) 및 10 개의 큰 데이터 쿼리 (L1-L10)로 이루어진다.

[0083] 실험에서는 큰 데이터 쿼리(L1-L10)를 제외한 모든 쿼리를 사용하였으며, L1 - L10을 생략 한 이유는 성능 평가에서 대부분의 엔진이 아직 이러한 쿼리를 실행할 수 없음을 확인 하였기 때문이다. [표 1]는 LargeRDFBench 쿼리 런타임과 본 발명에 따른 q-에러의 상관 관계를 보여준다.

[표 1]

Rank	1	2	3		4	5	6		
	Similarity Error				q-error				
Feature	E_J	E_P	E_T	Average	Q_J	Q_P	Q_T	Average	
FQ Engines	CostFed	0.23	0.59	0.43	0.42	0.14	0.26	0.1	0.17
	SemaGrow	0.33	0.33	0.33	0.33	0.47	0.37	0.001	0.28
	ODYSSEY	0.11	0.14	0.55	0.26	0.01	0.03	-0.06	-0.01
	SPLendid	0.3	0.4	0.24	0.32	0.17	0.1	0.24	0.17
	LHD	0.16	0.28	-0.2	0.08	0.51	0.11	0.04	0.22
	Average	0.22	0.35	0.27	0.28	0.26	0.17	0.06	0.17

[0088] 전체 쿼리 계획(EP)의 유사성 에러는 쿼리 실행 시간에 가장 큰 영향을 미치고(즉 0.35), 트리플 패턴(TP1, TP2 및 TP3)의 전체 트리플 패턴 에러(즉, ET 0.27), 조인의 Q-에러(즉, 0.26의 QJ), 조인 에러(즉 0.22의 EJ), 전체 계획의 Q-에러(즉, 0.17의 QP)와 삼중 패턴의 q-에러(즉, 0.06의 QT)이다.

[0089] 그 결과를 공평하게 비교하기 위해서, 본 발명에서는 모든 시스템이 통과한 공통의 질의만을 취한다. 본 발명에서는 20/32 벤치마크 쿼리에서 실패했기 때문에 LHD를 제거한다(매우 높은 수이고 12개의 간단한 쿼리만 통과됨). 그리고 비교에 적합하지 않다. 본 발명에서는 공통 질의에 Spearman의 상관관계를 다시 적용한다.

[0090] [표 2]은 제안된 방법이 일반적인 쿼리만 처리할 때 쿼리 런타임과 더 강력한 양의 상관 관계를 갖는다는 것을 보여준다.

[표 2]

Rank	Feature	1	2	3	4				5	6
		Similarity Error				q-error				
		E_J	E_P	E_T	Average	Q_J	Q_P	Q_T	Average	
FQ Engines	CostFed	0.54	0.61	0.36	0.5	0.11	0.23	0.05	0.13	
	SemaGrow	0.44	0.56	0.43	0.48	0.49	0.40	-0.02	0.29	
	ODYSSEY	0.22	0.42	0.53	0.39	-0.04	-0.01	-0.20	-0.08	
	SPLENDID	0.35	0.45	0.27	0.36	0.12	0.04	0.21	0.12	
	Average	0.39	0.51	0.40	0.43	0.17	0.17	0.01	0.12	

[0095] 쿼리 계획 에러(EP)의 유사성 에러 및 트리플 패턴(ET)은 쿼리 런타임에 가장 큰 영향(즉, 0.40)을 나타내며 조인 에러(즉, 0.39의 EJ)가 뒤따른다. 조인의 q-에러(즉 0.17의 QJ), 전체 질의 계획(즉 0.17의 QP), 3중 패턴의 q-에러(즉 0.01의 QT)이다.

[0096] 또한, 강력한 회귀 분석을 적용하여 결과에 영향을 미치는 특이치를 제거한다. 반복된 최소 제곱(IRLS)[37]에 의해 강력한 회귀 분석이 수행된다.

[0097] 본 발명에서는 무게 측정 기능으로 Huber 웨이트를 사용했다. IRLS에서 이 접근방식은 결과를 더욱 미세하게 조정했고, 우리가 제안한 유사성 에러와 런타임에 대한 상관관계를 더 강하게 만들었다.

[0099] [표 3]

Rank	1				2				3				4				5				6			
	Similarity Error								q-error															
	Feature	E_J	E_P	E_T	Average	Q_J	Q_P	Q_T	Average	Q_J	Q_P	Q_T	Average	Q_J	Q_P	Q_T	Average	Q_J	Q_P	Q_T	Average			
FQ Engines	CostFed	0.60	0.66	0.62	0.63	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16			
	SemaGrow	0.56	0.56	0.57	0.56	0.60	0.53	0.57	0.56	0.60	0.53	0.57	0.56	0.60	0.53	0.57	0.56	0.60	0.53	0.57	0.56			
	ODYSSEY	0.25	0.45	0.59	0.43	-0.04	-0.02	-0.20	-0.08	-0.04	-0.02	-0.20	-0.08	-0.04	-0.02	-0.20	-0.08	-0.04	-0.02	-0.20	-0.08			
	SPLendid	0.49	0.55	0.20	0.38	0.14	0.041	0.18	0.12	0.14	0.041	0.18	0.12	0.14	0.041	0.18	0.12	0.14	0.041	0.18	0.12			
	Average	0.45	0.56	0.49	0.50	0.22	0.18	0.18	0.19	0.22	0.18	0.18	0.19	0.22	0.18	0.18	0.19	0.22	0.18	0.18	0.19			

[0100]

[0102] [표 3]는 모든 메트릭스가 더 강력한 양의 상관 관계를 가지고 있음을 보여준다. 그러나, 본 발명에 따른 미터법에서 이 차이는 확실하다.

[0103] 전체 쿼리 계획(EP)의 유사성 에러는 쿼리 실행 시간에 가장 큰 영향(즉 0.56)을 나타내며, 트리플 패턴의 유사성 에러(0.49의 ET), 조인 에러(0.45의 EJ), 조인의 Q 에러(즉 0.22의 QJ), 전체 계획(즉, 0.18의 QP)과 트리플 패턴(즉, 0.18의 QP)의 Q-에러이다.

[0104] [표 3]는 또한 Odyssey에 대한 q-에러가 런타임과 부정적으로 상관되어 있음을 보여준다. 우리는 또한 그림 4의 높은 q-에러 값을 관찰할 수 있다. 전반적으로, 결과는 제안된 유사성 에러가 Q-에러보다 쿼리 런타임과 더 잘 상관된다는 것을 보여준다.

[0106] 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 이는 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다. 따라서, 본 발명 사상은 아래에 기재된 특허청구범위에 의해서만 파악되어야 하고, 이의 균등 또는 등가적 변형 모두는 본 발명 사상의 범주에 속한다고 할 것이다.

부호의 설명

[0108] 100: SPARQL 쿼리 처리 엔진 평가 장치

110: 트리플 패턴 에러 생성부

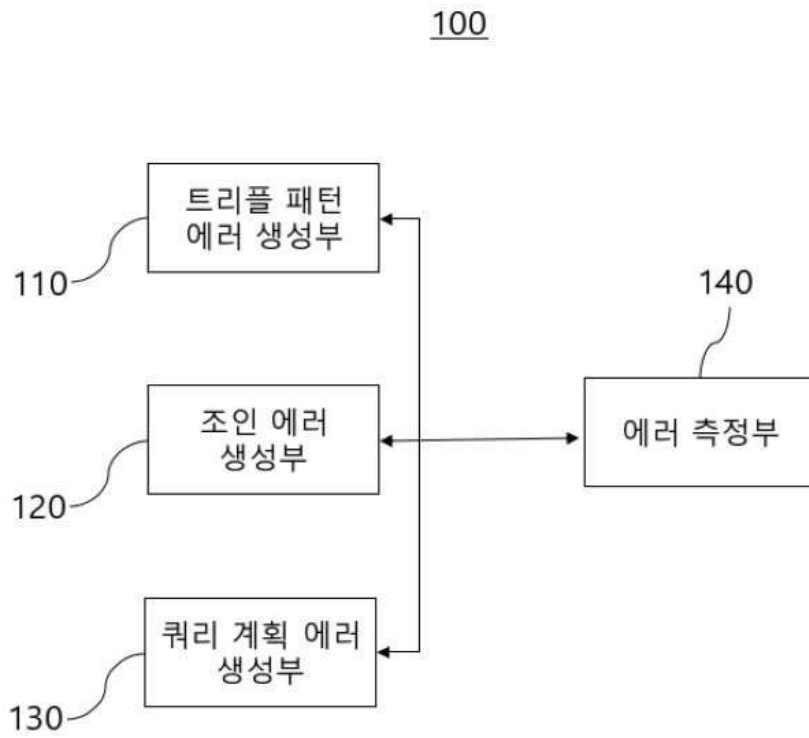
120: 조인 에러 생성부

130: 쿼리 계획 에러 생성부

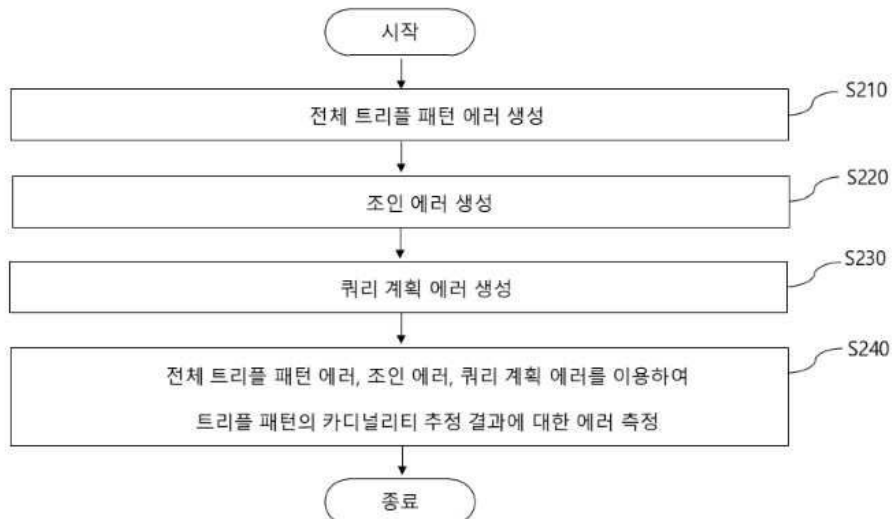
140: 에러 측정부

도면

도면1



도면2



도면3

```

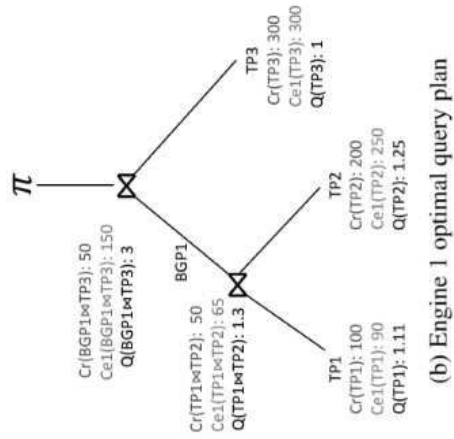
SELECT * WHERE {
  ?s :p1 ?o1 .
  Cr(TP1):100
  Ce1(TP1):90
  Ce2(TP1):200

  ?s :p2 ?o2 .
  Cr(TP2):200
  Ce1(TP2):250
  Ce2(TP2):600

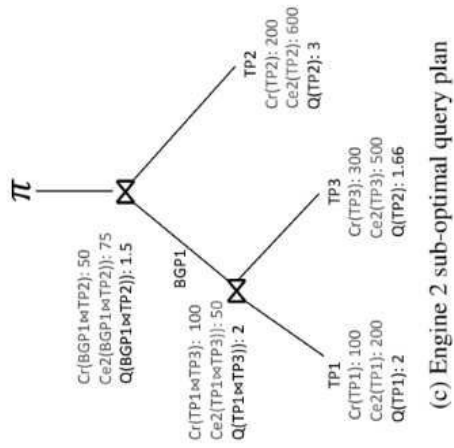
  ?s :p3 ?o3 .
  Cr(TP3):300
  Ce1(TP3):300
  Ce2(TP3):500 }

```

(a) Example query



(b) Engine 1 optimal query plan



(c) Engine 2 sub-optimal query plan