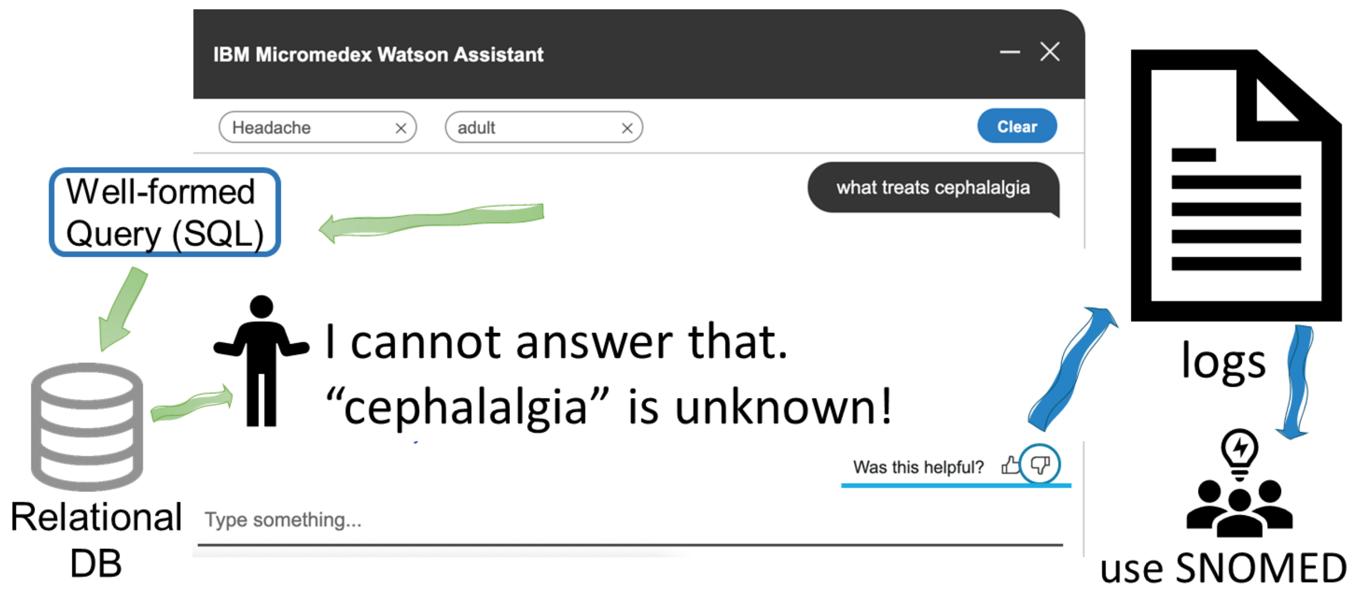
Ontology-Enriched Query Answering on Relational Databases

Shqiponja Ahmetaj, Vasilis Efthymiou, Ronald Fagin, Phokion G. Kolaitis, Chuan Lei, Fatma Özcan, Lucian Popa

IBM Research – Almaden, USA

Motivation


IBM Micromedex Watson Assistant

Headache

adult

what treats cephalalgia

I cannot answer that.
“cephalalgia” is unknown!

Was this helpful?

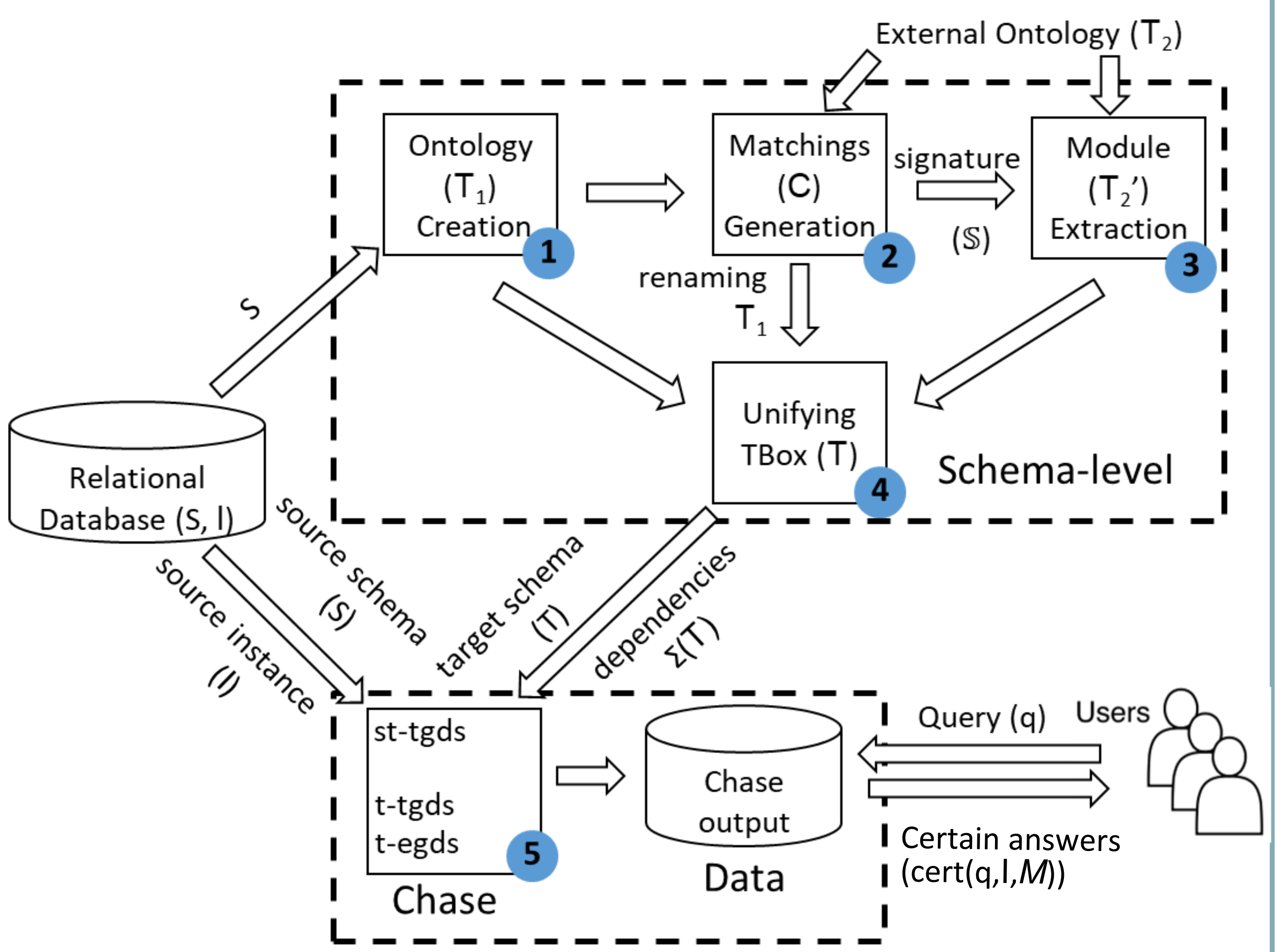
Type something...

A. Quamar, C. Lei, D. Miller, F. Özcan, J. Kreulen, R. Moore, V. Efthymiou. An Ontology-Based Conversation System for Knowledge Bases. *SIGMOD 2020*

Main Challenges

- Identify and reuse only the parts of SNOMED CT that are relevant
  - used existing tools from different AI communities
    - ontology creation from a relational DB, ontology matching, module extraction
  - designed a flexible framework that goes beyond our use case (github.com/IBM/ontology-enriched-query-answering)
- Answer queries expressed over the vocabulary of SNOMED CT using our database
  - Two main approaches exist:
    - Materialization:
      - » Materialize a universal solution (once) using the *chase* procedure
      - » Use the universal solution to compute the certain answers on arbitrary CQs over the target schema
    - Query Rewriting: Keep the original data, but rewrite user query when it is submitted before evaluating it

Framework Architecture



1 Ontology Creation

- Generate an ontology T$_1$ from a given relational DB
  - includes functionality, domain and range

2 Matchings Generation

- Find matchings between the generated ontology T$_1$ and an external ontology T$_2$ (e.g., SNOMED CT)

3 Module Extraction

- Retain a small subset (T’$_2$) of the external ontology that is relevant to a signature of interest 𝕊
- We determine 𝕊 based one the matchings of Step 2

4 Unifying the TBox

- Merge the ontology T$_1$ with the 𝕊 -module (T’$_2$)
- for merged terms, use the names specified in T’$_2$

5 Query Answering via the Chase

- Use input relational schema as source schema S
- Use the unified TBox as target schema T
- Generate st-tgds, t-tgds, and t-egds from S and T
  - Run the chase, and compute the certain answers

Chase Termination

Expressivity of the Unified TBox: acyclic $\mathcal{ELH}^{fdr}$

- $\mathcal{ELH}$ extended with domain and range, and limited functionality:
  - functional roles not allowed on the RHS of axioms
- Classic acyclicity definition extended with additional conditions for domain, range and functionality
  - E.g.: $A \sqsubseteq \exists r,\ rng(r) \sqsubseteq A$ (acyclic under the $\mathcal{ELH}$ acyclicity conditions, but results in infinite chase)

st-tgds: From every relation R of the source schema S

$$R(\underline{x_1}, \dots, x_n) \to R'(x_1) \wedge R'^{1,2}(x_1, x_2) \wedge \cdots \wedge R'^{1,n}(x_1, x_n),$$

where x_1 is the primary key of R, and R’, $R'^{1,j}$ are fresh relation names.
If (R, R”) ∈ C, we replace $R'(x_1)$ above with $R''(x_1)$.

t-egds: Every functional role r gives rise to the t-egd

$$r(x,y) \wedge r(x,z) \to y = z$$

t-tgds: For every $\mathcal{EL}$ concept C, there is a CQ $q_C(x)$ with a free variable x, s.t. $C(x) \equiv q_C(x)$

- **Case 1:** $q_C(x) \coloneqq \exists \bar{y} \varphi_C(\bar{y}, x)$,
  where $\bar{y}$ is a non-empty tuple of variables
- **Case 2:** $q_C(x) \coloneqq A_1(x) \wedge \cdots \wedge A_n(x)$,
  where $A_1(x), \dots,\ A_n(x)$ are concept names

The tgds arising from an $\mathcal{ELH}^{fdr}$ terminology have one of the following 7 types:

1) $A(x) \to \exists \bar{y} \varphi_C(\bar{y}, x)$ *($A \sqsubseteq C$, where C is of Case 1)*
2) $A(x) \to A_1(x) \wedge \cdots \wedge A_n(x)$ *($A \sqsubseteq C$, where C is of Case 2)*
3) $\varphi_C(\bar{y}, x) \to A(x)$ *($C \sqsubseteq A$, where C is of Case 1)*
4) $A_1(x) \wedge \cdots \wedge A_n(x) \to A(x)$ *($C \sqsubseteq A$, where C is of Case 2)*
5) $r_1(x,y) \to r_2(x,y)$ *($r_1 \sqsubseteq r_2$)*
6) $r(x,y) \to A(x)$ *($dom(r) \sqsubseteq A$)*
7) $r(x,y) \to A(y)$ *($rng(r) \sqsubseteq A$)*

Theorem: Let T be an acyclic $\mathcal{ELH}^{fdr}$ terminology and let $\Sigma(T)$ be the associated set of tgds and egds.
Then $\Sigma(T)$ is C-stratified.

Experimental Evaluation

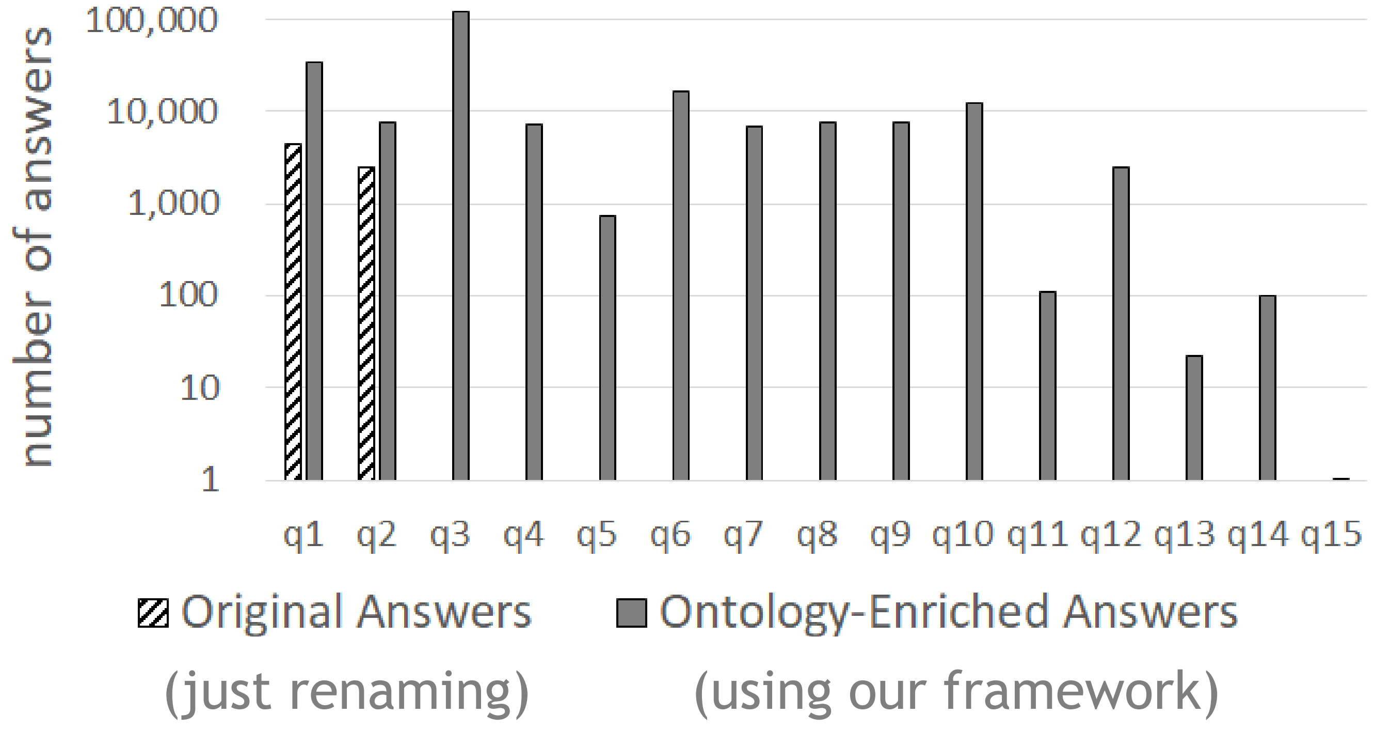
Input DB (MDB): 62 relations, 158FKs, 500k+ tuples

Ontologies:

- MDB ontology: 49 concepts, 170 roles (156 funct.)
- SNOMED CT: 356k concepts, 119 roles (0 funct.)
- 12 matchings identified (i.e, 12 elements in 𝕊)
- 𝕊 -module in SNOMED: 35 concepts, 7 roles
- Unified TBox: 72 concepts, 177 roles (156 funct, 170 with domain and range)

Chase: 62 st-tgds, 154 t-tgds, 156 t-egds

15 queries selected from system logs Jan-Jun 2019



Chase execution time: **1,676ms** (done once, offline)

QA times: **64ms** on average (min: 1ms, max: 576ms)

Space overhead: from 62.3MB (source instance) to 77.5MB (chase output)

Our framework is beneficial for two types of queries:

- CQs whose conjuncts all appear in MDB, but we learned something new about them from SNOMED
- CQs with some conjuncts unknown
  - could not be answered originally

https://github.com/IBM/ontology-enriched-query-answering

S. Ahmetaj, V. Efthymiou, R. Fagin, P.G. Kolaitis, C. Lei, F. Özcan, L. Popa. Ontology-Enriched Query Answering on Relational Databases IAAI-21