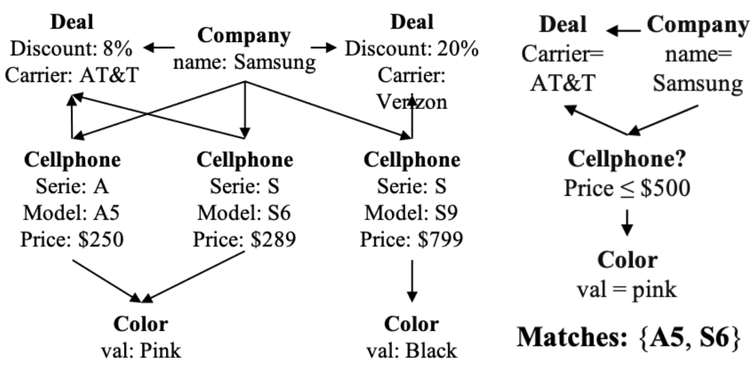
Answering Why-questions by Exemplars in Attributed Graphs

Mohammad Hossein Namaki*, Qi Song*, Yinghui Wu*, Shengqi Yang#
Washington State University* WeWork#
{m.namaki, qi.song, yinghui.wu}@wsu.edu* shengqi.yang@wework.com#

Introduction

- **Subgraph queries: access and understand complex networks.**
- **Writing queries is nevertheless a nontrivial task for end users:**
  - ✓ Users often need to revise the queries multiple times to find desirable answers.
  - ✓ It is easier to provide *examples of interest*

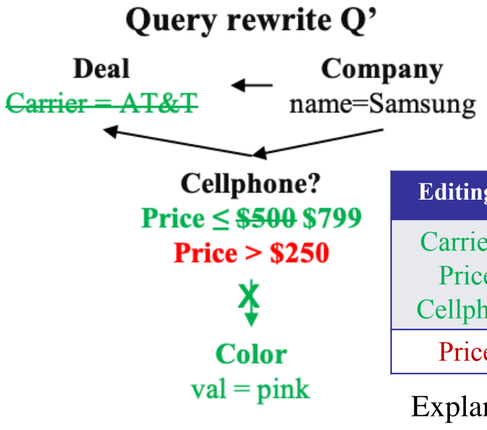
"Find Samsung cellphones packed with color pink and carrier AT&T, cheaper than $500"



- **Why-questions by exemplars**
  - ✓ Why question: "why some (unexpected) entities are in the query answer?"; and
  - ✓ Why-not question: "why certain entities are missing from the query result?"
  - ✓ Exemplars: entities or SQL queries

*"I want to see some cellphones which are similar to **S9**"*

Query rewrite Q'



Editing Operator	Matches
Carrier = AT&T Price ≤ $799 Cellphone-Color	S9
Price > $250	A5

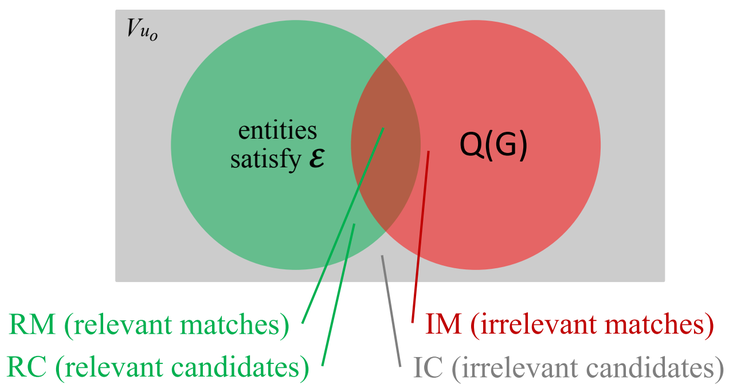
Explanation: a picky table

Matches: {S6, S9}

Answers for Why-Questions

- **Query rewrites: query editing operators**
  - ✓ Remove/Add edge/literal;
  - ✓ Relax/refine literal/edge bound
- **Answer closeness** $cl(Q'(G), \mathcal{E})$**:**

$$\frac{\sum_{v \in RM(\mathcal{E},Q)} cl(v,\mathcal{E}) - \lambda |IM(\mathcal{E},Q)|}{|V_{u_o}|}$$



RM={S6}, IM={A5}, RC={S9}, IC={}

Problem Formulation

Input: query Q, answer Q(G), graph G, a Why-question W with exemplar $\mathcal{E}$, and editing budget B

Output: a query rewrite $Q' = Q \oplus O^*$ such that

$$O^* = \arg\max_{O: c(O) \le B} cl(Q'(G), \mathcal{E})$$

Q-Chase Representation

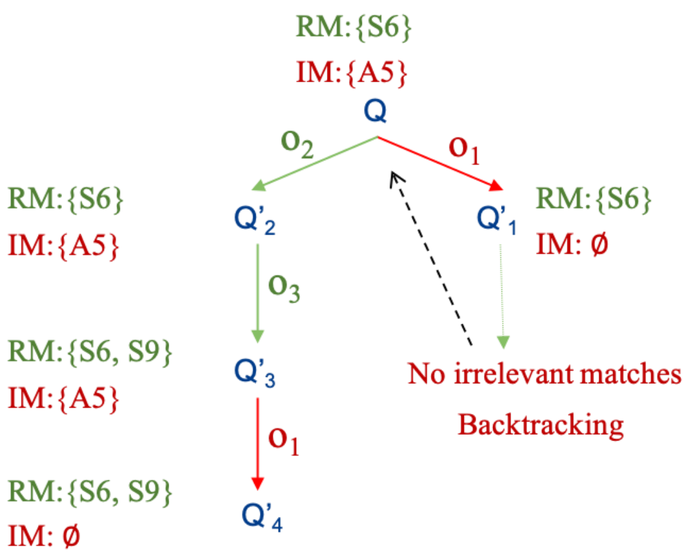
- Q-Chase enforces value constraints from $\mathcal{E}$ over subgraph queries and their answers in G
- It follows an inductive process
  - ✓ Initialization: $(Q_0, \mathcal{E}_0)=(Q, \mathcal{E})$
  - ✓ QChase step i: $(Q_i, \mathcal{E}_i) = \text{QChase}(Q_{i-1}, \mathcal{E}_{i-1})$

$$(Q_{i-1}, \mathcal{E}_{i-1}) \xrightarrow{\text{modifying } Q_{i-1} \text{ to } Q_i} (Q_i, \mathcal{E}_i)$$

- A Q-Chase is terminated when all the examples are satisfied ($\mathcal{E}_i=\emptyset$)

Answering Why-Questions

- **A best first search strategy with backtracking to simulate Q-Chase.**
  - ✓ Generate a set of "picky" operators;
  - ✓ Select the best operators;
  - ✓ Constructs a new Q' and evaluate cl().
  - ✓ Terminate: no budget or achieve optimal Q'

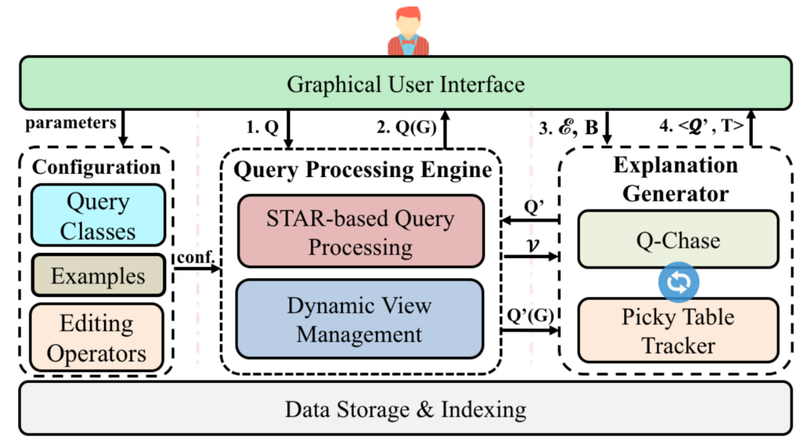


- **Anytime behavior**
- **Optimization:**
  - ✓ Closeness upper bound (pruning);
  - ✓ Ad-hoc star views: picky operator generation and incremental evaluation

Extensions

- **Answering Why-many Questions**
  - "Why this query has so many answers?"
  - ✓ Find a Q' that removes IMs
  - ✓ Approximation algorithm: $1/2(1 - 1/e)$
- **Answering Why-Empty Questions**
  - "Why this query has no answer?"
  - ✓ Find a Q' with at least one RM
  - ✓ Exact algo.: for star query and removal-only operators

NAVIGATE Architecture



Evaluation

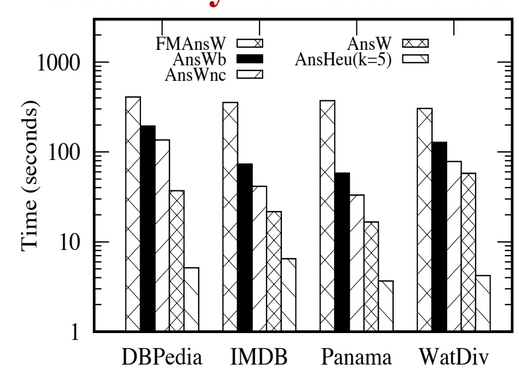
Datasets

Name	Description	# of nodes	# of edges	# of attributes per node
DBPedia	Knowledge Graph	4.86M	15M	9
IMDb	Movie Network	1.7M	5.2M	6
Offshore	Financial Activities	839K	3.6M	4
WatDiv	e-commerce information	521K	9.1M	8

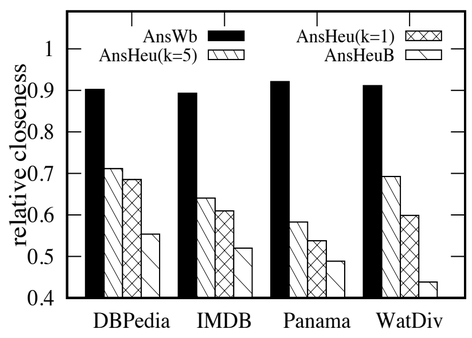
Algorithms

- **Exact algorithms**
  - ✓ AnsW: utilizing caching and pruning strategies
  - ✓ AnsWnc (no caching) & AnsWb (no pruning)
- **Heuristic algorithms:** AnsHeu & AnsHeuB
- **Baseline:** FMAnsW (pattern mining based)

Efficiency and Effectiveness



AnsW outperforms FMAnsW, AnsWb, and AnsWnc by 10.82, 3.41, and 2.1 times, respectively



AnsW constantly achieves the maximum closeness among others. AnsHeu can suggest good rewrites with relative closeness 66%.

Case analysis

- **User study:**
  - ✓ Ask users to re-rank the top-3 query rewrites from AnsW;
  - ✓ Good answer relevance: nDCG_3 = 0.71.
- **Case study:** A user searched for recent computer models with GPU (why-empty)
  - ✓ Desired laptops powered by either Intel or AMD GPU