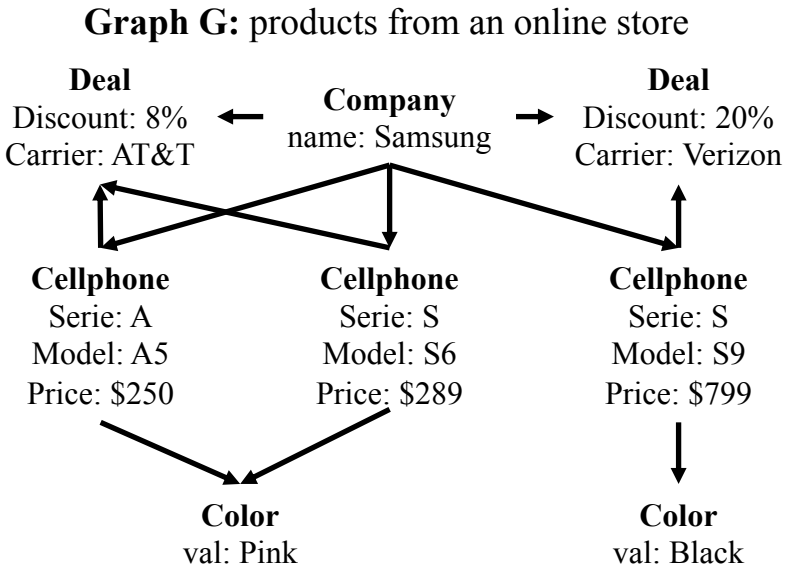
Answering Why-questions by Exemplars in Attributed Graphs

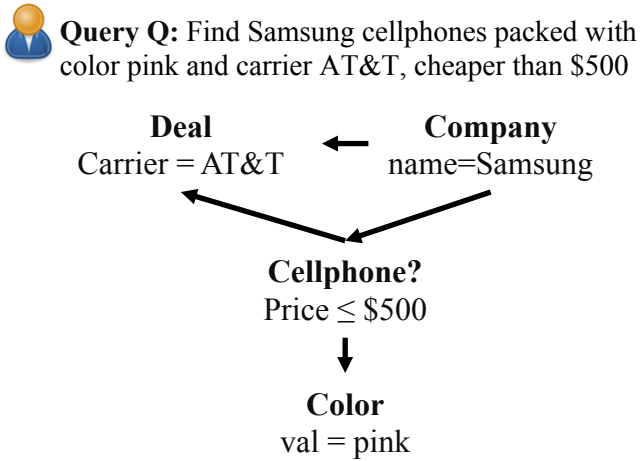
Mohammad Hossein Namaki[1] **Qi Song[1]** Yinghui Wu[1,2] Shengqi Yang[3]

1 Washington State University 2 Pacific Northwest National Laboratory 3 wework

Multi-attributed Graph and Subgraph Query

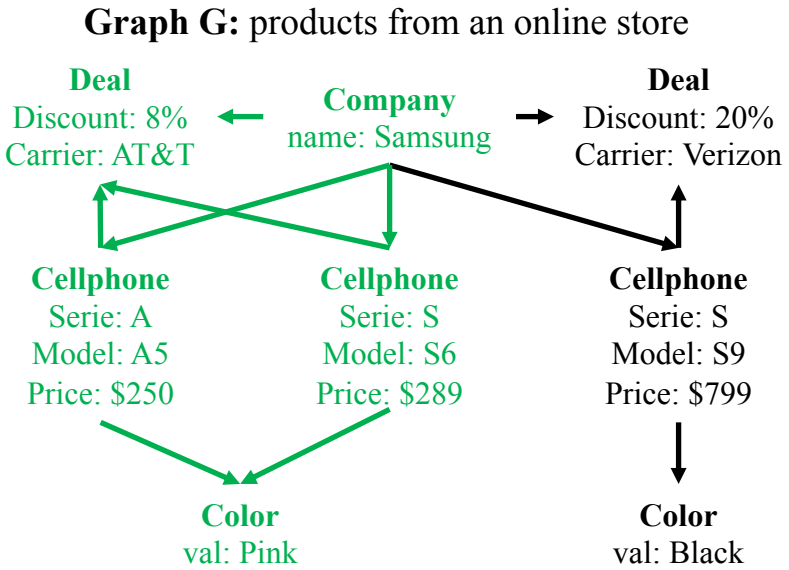
- Multi-attributed graphs: a labeled graph with attributes on nodes
- Subgraph query Q: a (labeled) graph pattern with a focus node u_o
- Knowledge extraction, POI recommendation, Medical Analysis, etc..

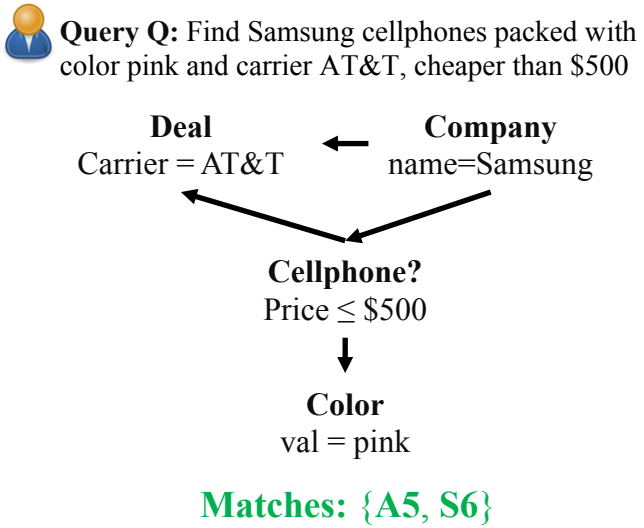




Multi-attributed Graph and Subgraph Query

- Multi-attributed graphs: a labeled graph with attributes on nodes
- Subgraph query Q: a (labeled) graph pattern with a focus node u_o
- Knowledge extraction, POI recommendation, Medical Analysis, etc..



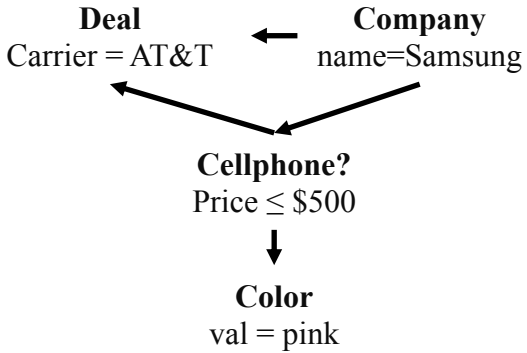


Why-questions by Exemplars

- Why-questions
  - **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: User specified entities / SQL statement to characterize answers

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

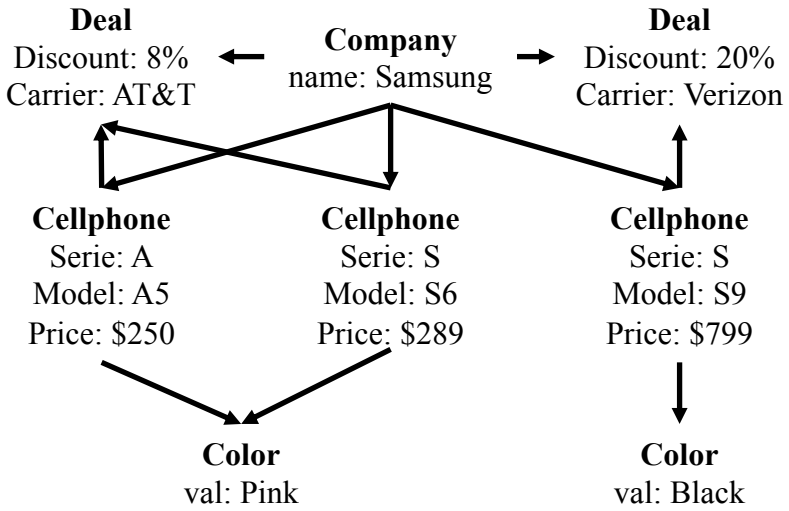
Relevant entities: {**S6, S9**}, Irrelevant entities: {**A5**}

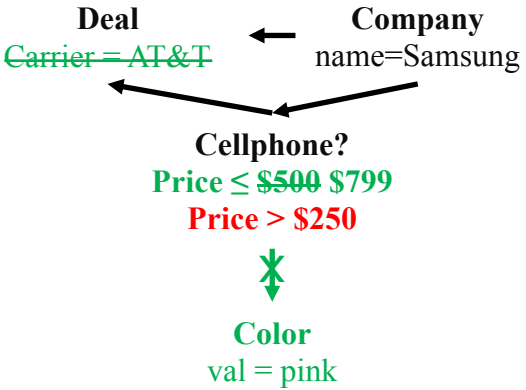


Why-questions by Exemplars: an Example

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

Graph G: products from an online store

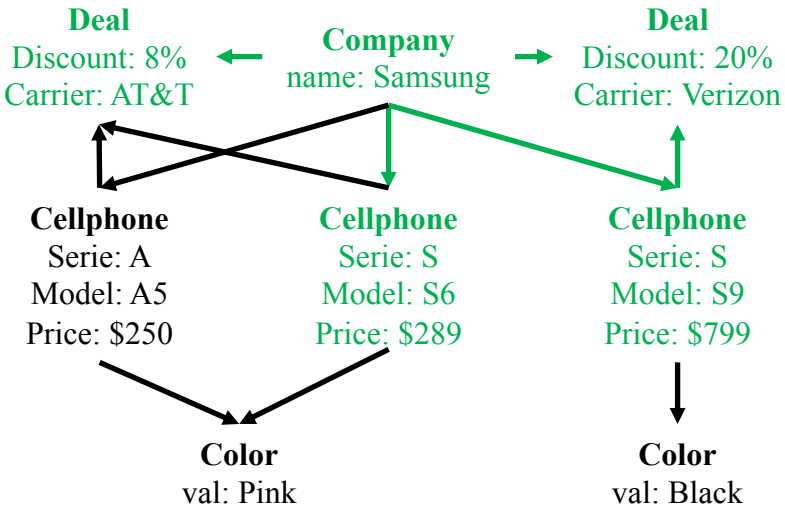




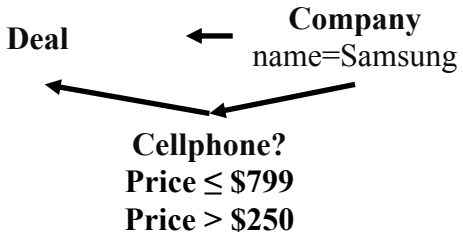
Why-questions by Exemplars: an Example

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

Graph G: products from an online store



Query rewrite Q'



Matches: {**S6**, **S9**}

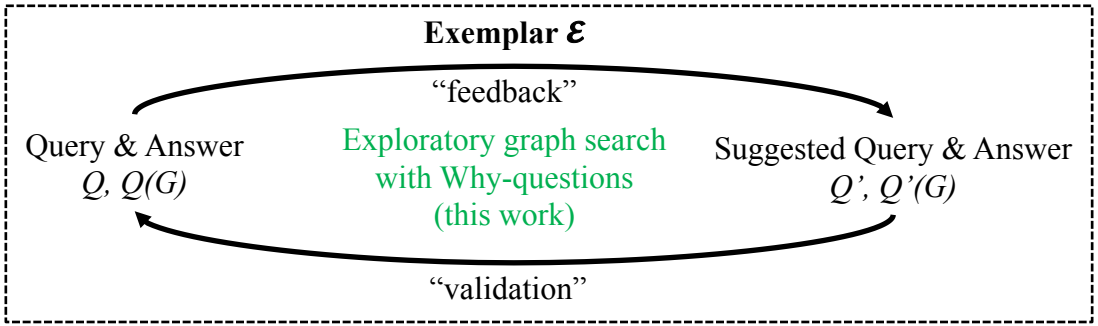
Editing Operator	Matches
Carrier = AT&T Price ≤ $799 Cell. - Color	S9
Price > $250	A5

Explanation

Graph Exploratory Search Guided by Examples

A new formalization of answering Why-questions by exemplars.

Q-Chase: a new symbolistic formalization.



Feasible algorithms with performance guarantees.

Application (demo): NAVIGATE

Answering Why-questions: Problem Formulation

- Why-question:

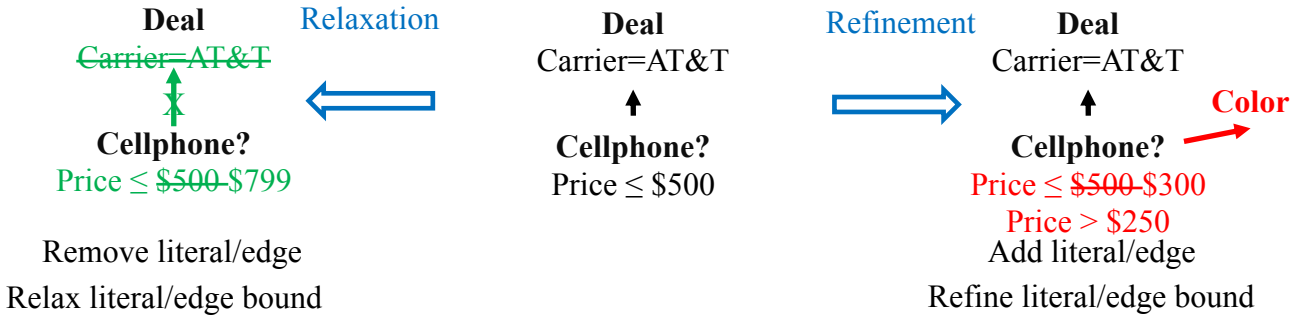
$$W(Q(u_o), \mathcal{E})$$

u_o: output node $Q(u_o, G)$: answer of Q $\mathcal{E}$: exemplars

$\mathcal{E}$=(T,C), where T is a table with all attributes from the graph, C is a conjunction of conditions (SQL queries)

rep($\mathcal{E}$, V): nodes in V that satisfies the constraints in $\mathcal{E}$

- Answer of Why-questions: query rewrites (operators with editing cost)

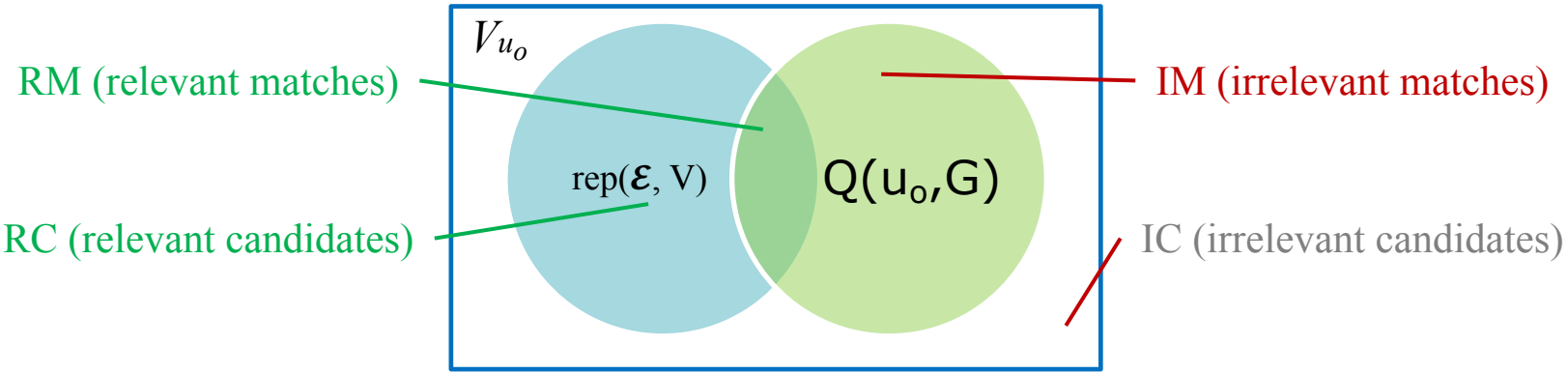


Answering Why-questions: Quality Measurements

- Answer closeness:

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

Rewarded and penalized by weighted number of RM and the number of IM in the answer set



Answering Why-questions: Problem Statement

- **Input:** a query Q, answer $\mathrm{Q}(G)$, graph G, a Why-question W, and editing budget B,
- **Output:** a query rewrite $\mathrm{Q}' = \mathrm{Q} \oplus O^*$, such that

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

- The problem of answering Why question by exemplar is **NP-hard** and **hard to approximate**.
- Solution overview:

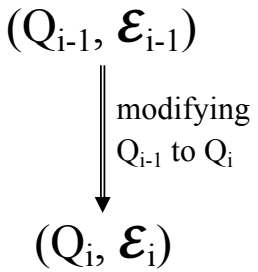
Problem	Property	Algorithm	Description
Why-Question	NP-Hard Not in APX	AnsW	Exact
		AnsHeu	Heuristic
Why-Many	Approximable	ApxWhyM	$1/2(1 - 1/e)$ approximation
Why-Empty	P-Time for special case	AnsWE	Exact for special case

A Characterization by Extending Chase

- Q-Chase enforces value constraints from $\boldsymbol{\mathcal{E}}$ over subgraph queries and their answers in G

In contract to conventional Chase

- An inductive process
  - Initialization: $(Q_0, \mathcal{E}_0)=(Q, \mathcal{E})$
  - Q-Chase step i: $(Q_i, \mathcal{E}_i) = Q\text{-}Chase(Q_{i-1}, \mathcal{E}_{i-1})$



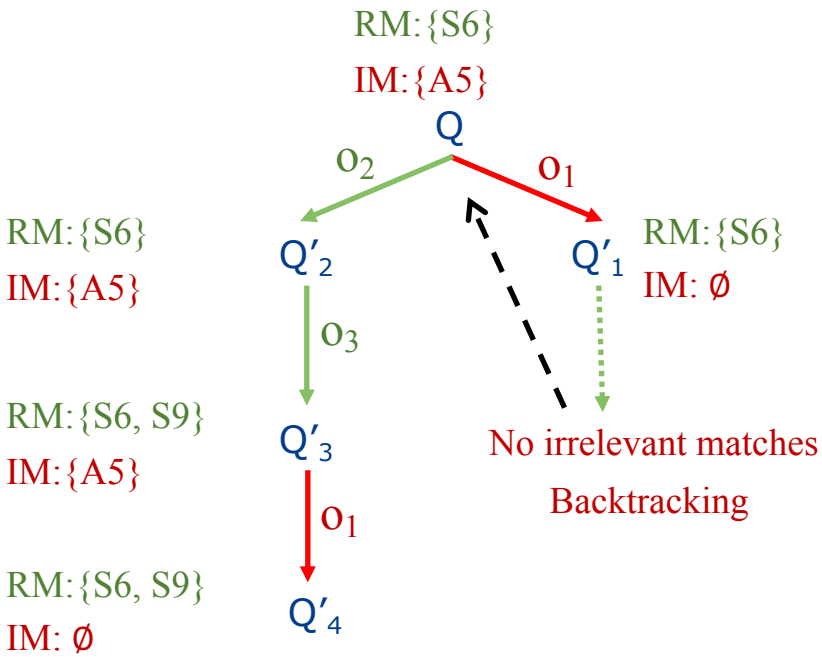
- A Q-Chase is terminated when all the constraints in $\boldsymbol{\mathcal{E}}$ are satisfied

- Finding the optimal terminal sequence that maximizes answer closeness.

Computing Optimal Query Rewrite - Algorithm

- Our idea: performing a best first search strategy with backtracking to simulate Q-Chase
  - Generate and prioritize a set of "picky" operators;
  - Query construction and evaluation;
  - Terminate when no budget or achieve optimal Q'.

operators that are likely to improve closeness



- Performance analysis:
  - Anytime behavior
  - Delay time to simulate a Q-Chase:

$$O(|Q.S(G)|_{b_m}^{|Q|+1})$$

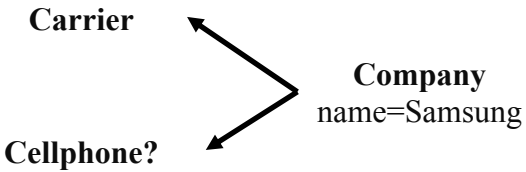
$|Q.S(G)|_{b_m}$: b_m-hop neighbors of the edges in star views in G

b_m: maximum edge bound

Computing Optimal Query Rewrite - Optimization

- Pruning: closeness upper bound
- Star views: a primitive structure
  - Query decomposition
  - Picky operator suggestion
  - Incremental query evaluation

Star query Q_{11}



Price ≥ 790 RAM ≥ 4
Display ≤ 6.4 Storage ≥ 64

RM = {P_3,P_4,P_5}
IM = {P_1,P_2}

Star table T_{11}

Company	Cell-Carrier	Cell-Company
P_1	AT&T	Samsung
P_2	T-Mobile	Samsung
P_3	Sprint	Samsung
P_4	Sprint	Samsung
P_5	Sprint	Samsung

AddL(Carrier.name = 'Sprint') Picky

AddL(Carrier.name = 'AT&T')

Experimental Setting

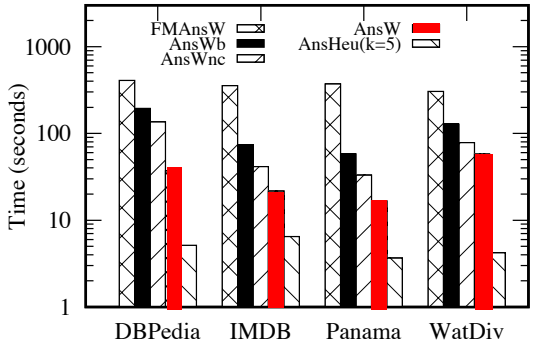
- Datasets

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 (synthetic)	521K	9.1M	8

- Ground truth queries: benchmark queries and a query generator.
- Question generation: disturb *(Q*, Q*(G))* by injecting random atomic operators.
- Algorithms
  - **Exact algorithms**: AnsW (optimized), AnsWnc (no caching), AnsWb (no pruning).
  - **Heuristics**: AnsHeu, and AnsHeuB (random).
  - **Baseline:** FMAnsW (Mottin et. al. 2015).

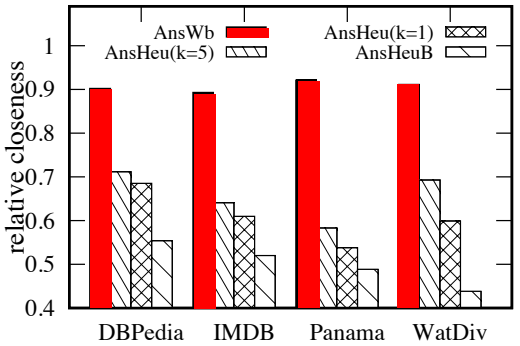
Experimental Result

- Answering Why questions: efficiency and effectiveness

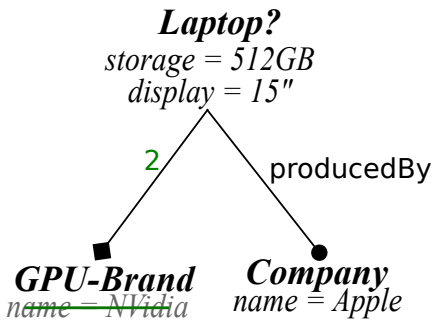


AnsW outperforms FMAnsW, AnsWb, and AnsWnc by 10.82, 3.41, and 2.1 times, respectively

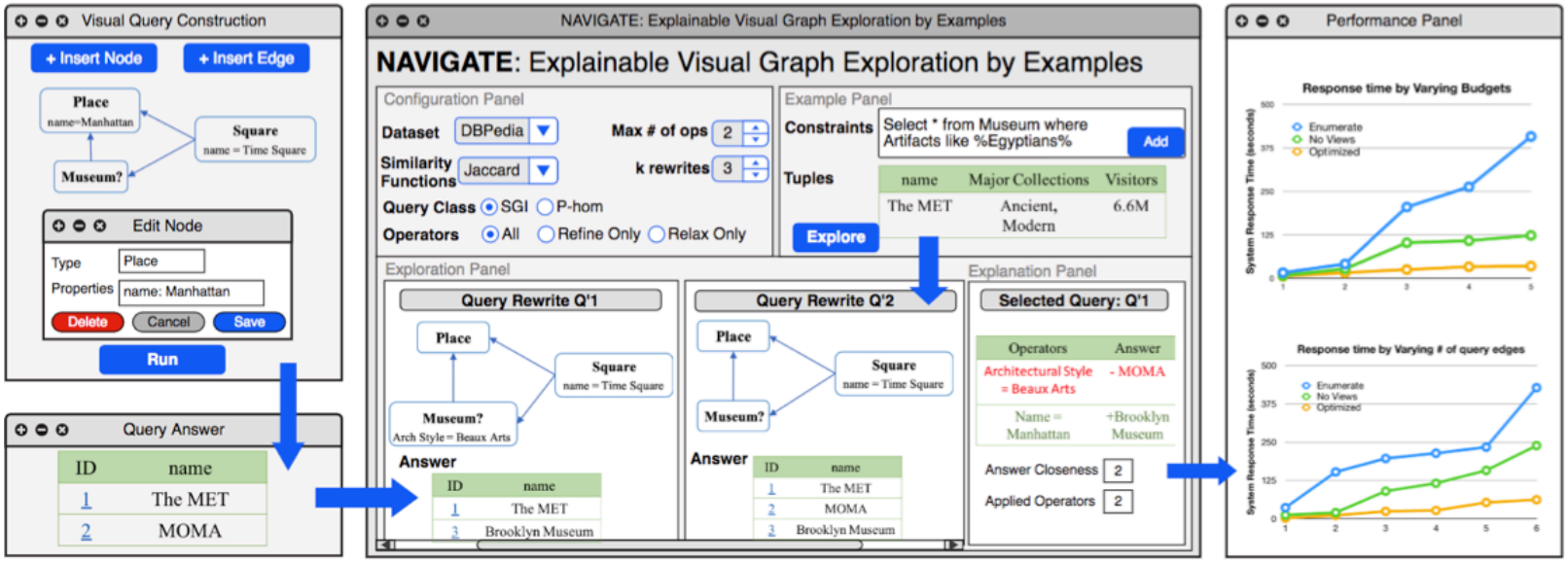
AnsW constantly achieves the maximum closeness among others.



- User study:
  - Ask users to re-rank the top-3 query rewrites from AnsW;
  - Good answer relevance: nDCG_3 = 0.71.
- Case analysis: product recommendation
  - A user searched for recent computer models with GPU (after 2018)
  - Desired laptops powered by either Intel or AMD GPU



A Demo system



NAVIGATE: Explainable Visual Graph Exploration by Examples

Wednesday & Thursday 16:20 – 17:50

Thank you!