Multi-Example Search in Rich Information Graphs

Matteo Lissandrini, *Davide Mottin, Themis Palpanas, Yannis Velegrakis*

db Trento

What are you looking for?

dbTrento

Search by **a list** of **specifications**

1. A **Movie** and an **Actor**
2. From the **Movie** return the **Director**
3. The **Director** has won an **Award**
4. The **Movie** is adapted from a **Book**
5. From the **Book** return the **Author**

Hard to Specify!

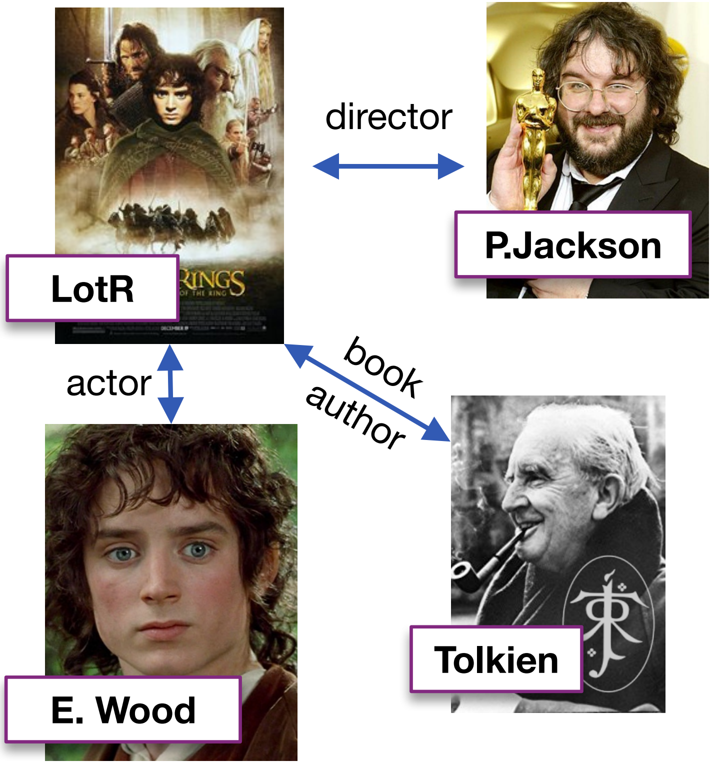
Too many options!
What is the schema?
Which specification is important?

Search by **Example**

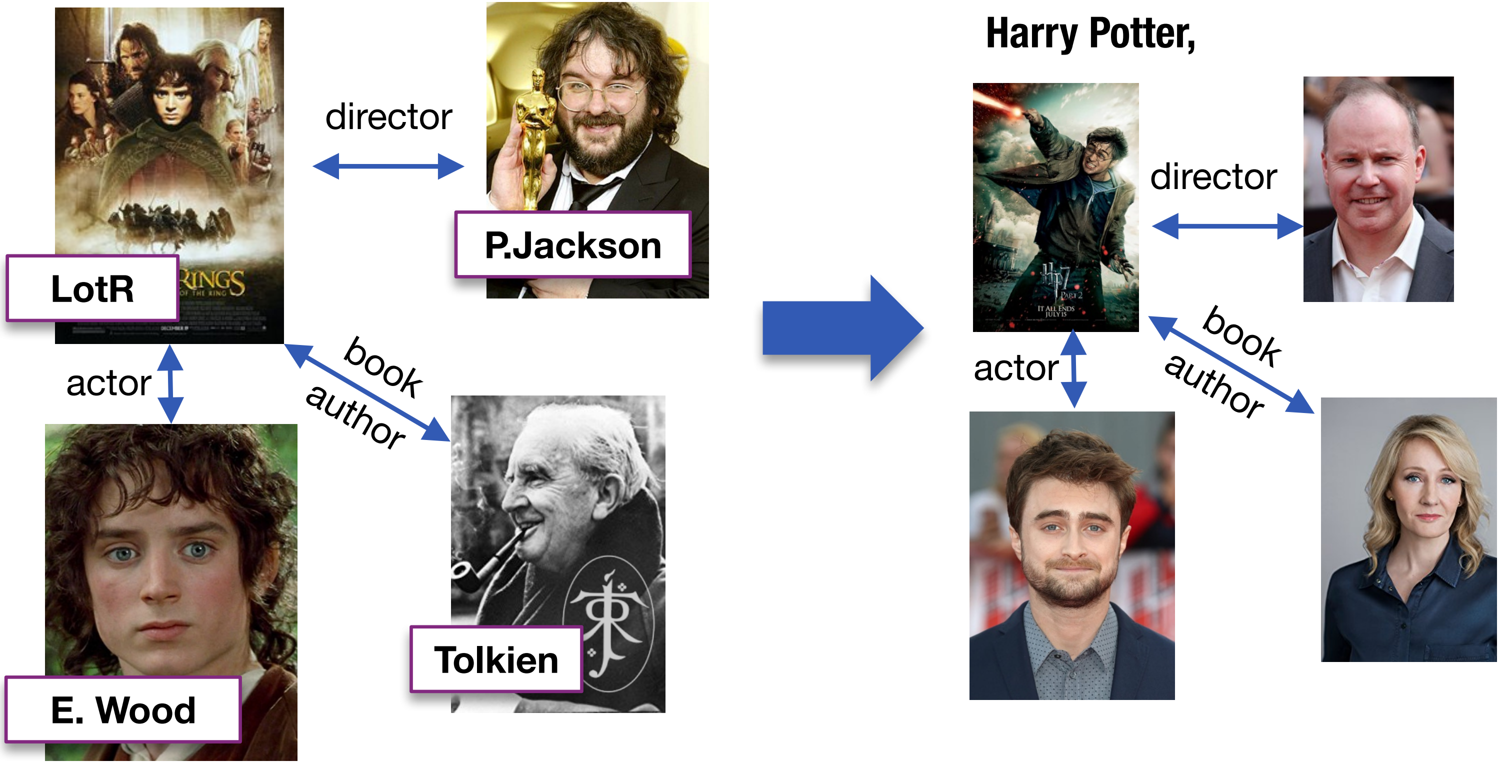
Lord of the Rings | E. Wood | P.Jackson | Tolkien

dbTrento

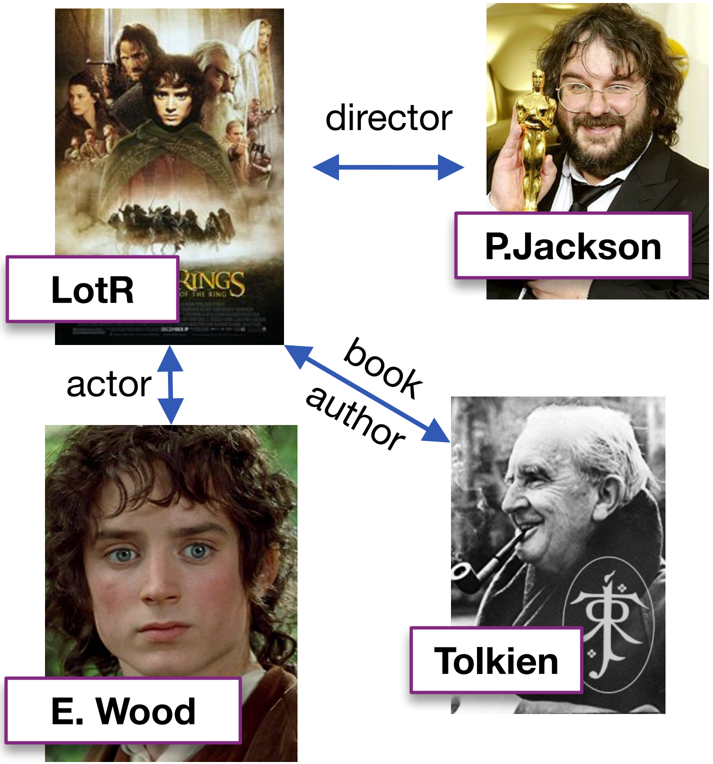
Search by **Example**



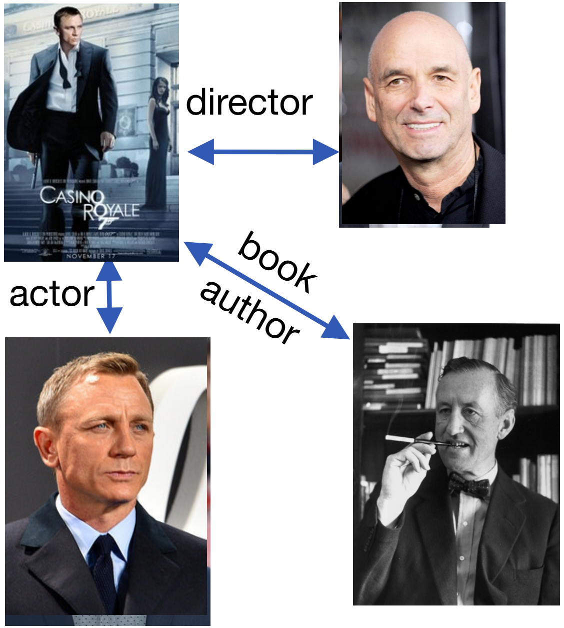
Search by **Example**



Search by **Example**

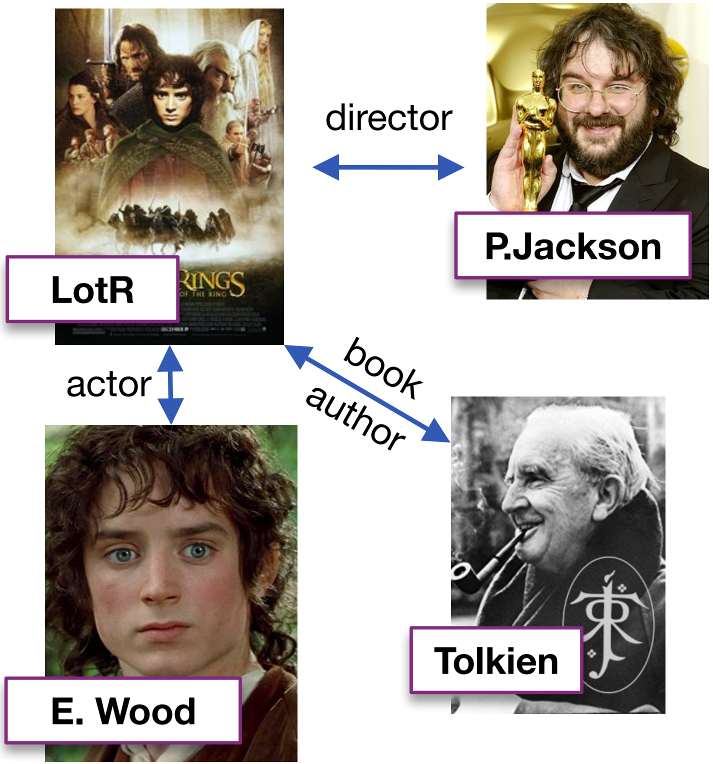


Harry Potter, James Bond, …

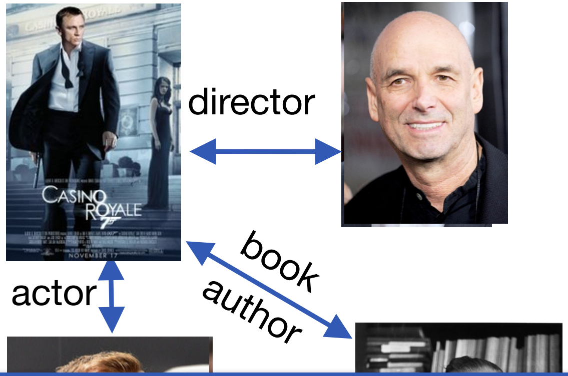


db Trento

Search by **Example**



Harry Potter, James Bond, …



More Intuitive!

Avoid list of specifications

dbTrento

ONE EXAMPLE IS NOT ENOUGH

When
known examples
are only
Partial Specifications

db Trento

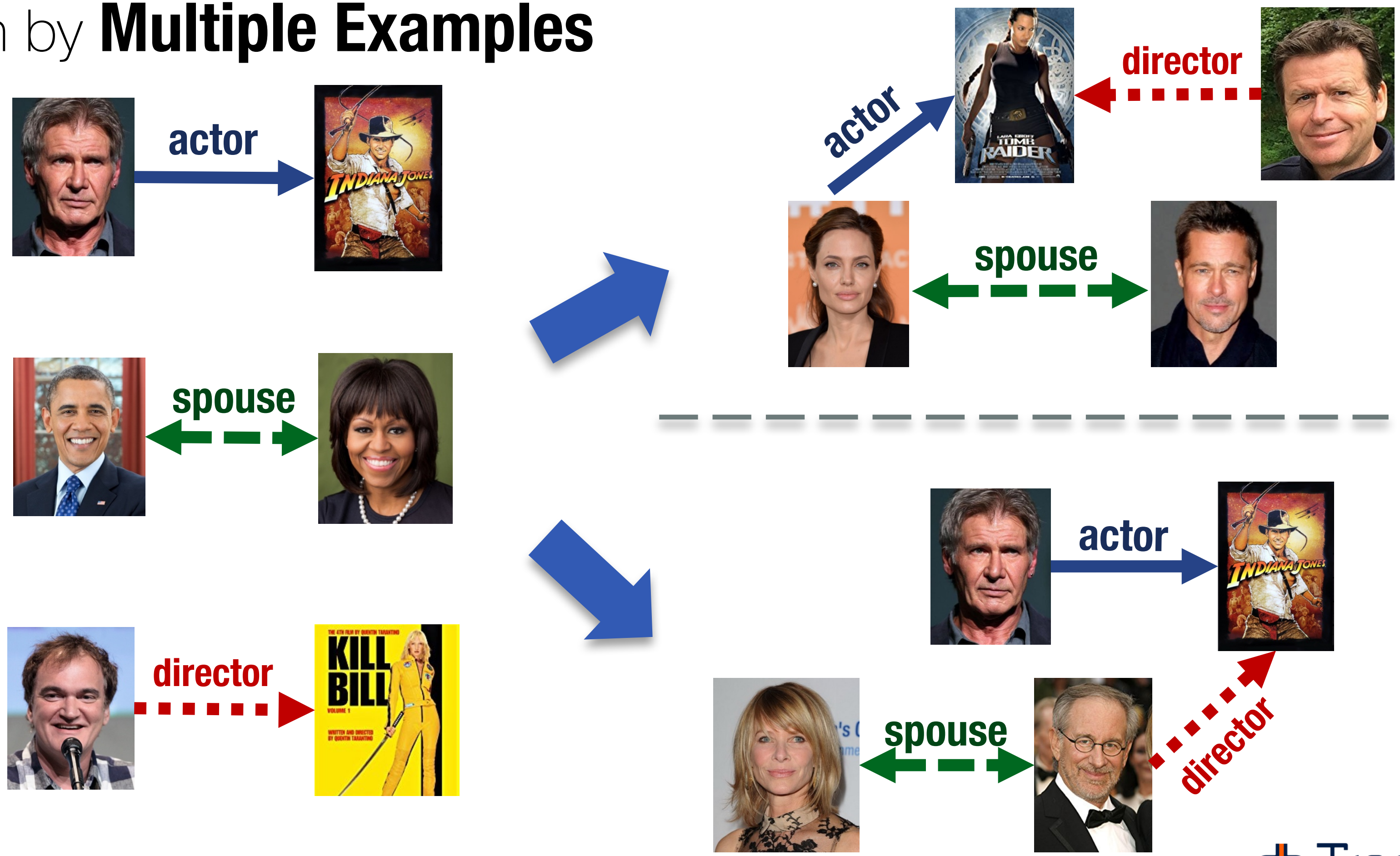
Search by **Multiple Examples**

ONE EXAMPLE IS NOT ENOUGH

When
Results have
<u>Different Structures</u>

db Trento

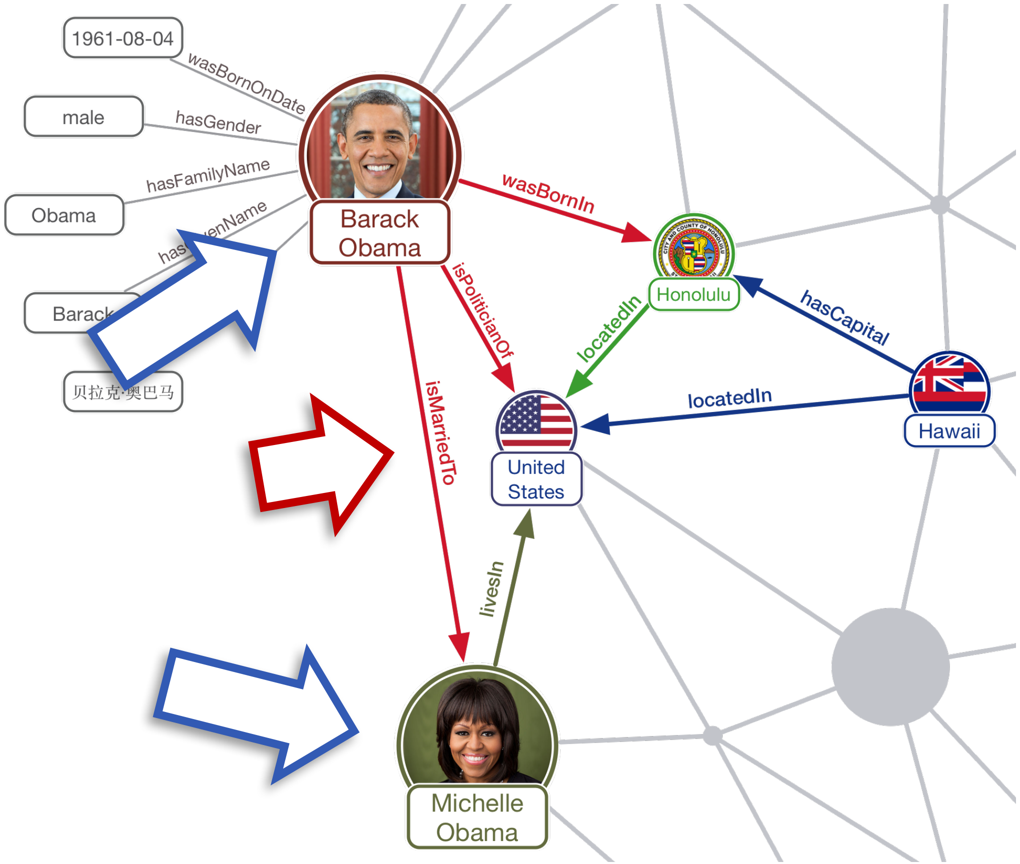
Search by **Multiple Examples**



Multi-Example Search

- Multiple Simple Examples
- Each Example describes an Aspect
- Results are Combinations of aspects
- Results have possibly Multiple Structures

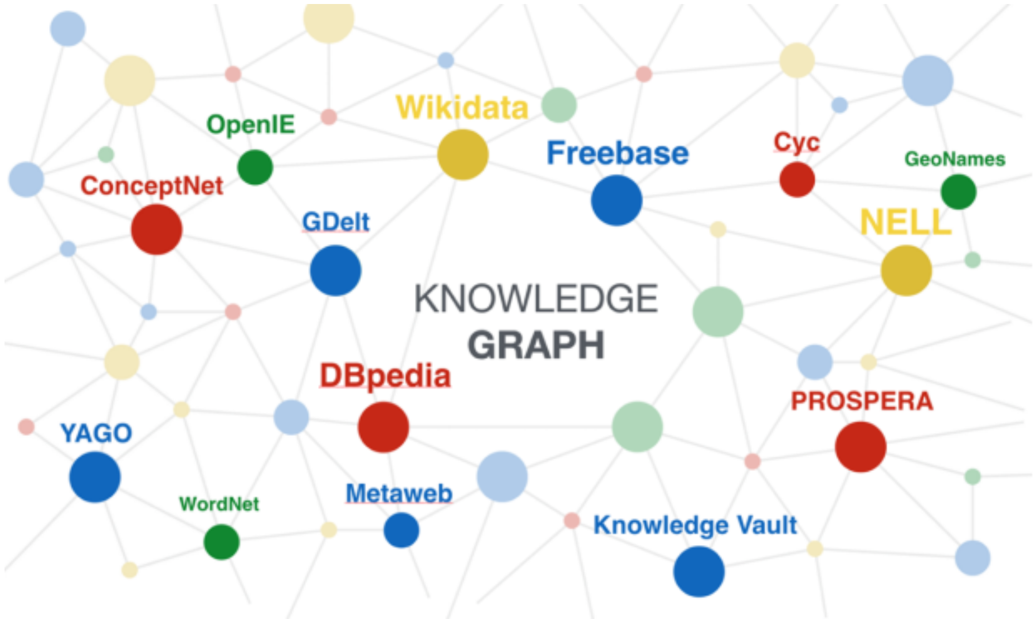
INFORMATION GRAPHS



Nodes & Edges

Edge-labelled
Multigraphs

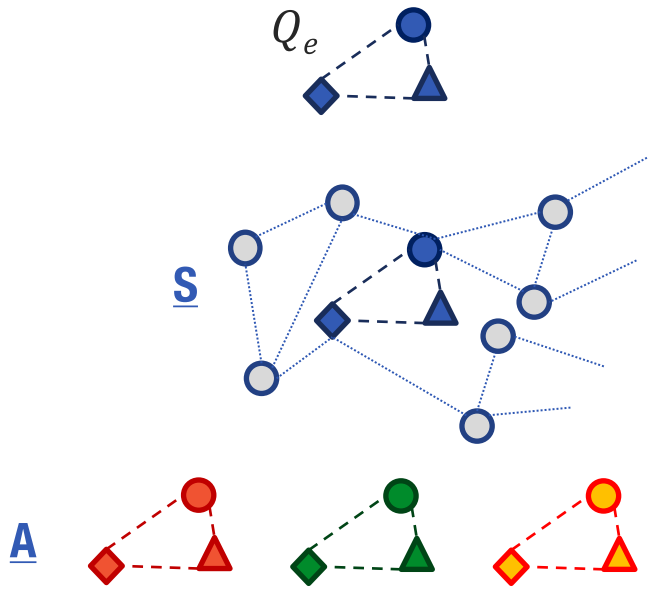
$$G: \langle V, E, L, \ell \rangle$$



dbTrento

SINGLE EXAMPLE

Exemplar Queries



Input: Q_e, an example ***element*** of interest

Output: set of elements in the ***desired result set***

Exemplar Query Evaluation

- **match** Q_e to ***sample S*** in the graph G
- **find** the set of ***elements A*** similar to S given a ***similarity relation***

- [OPTIONAL] return only the ***top-k subset*** $A^K \subseteq A$

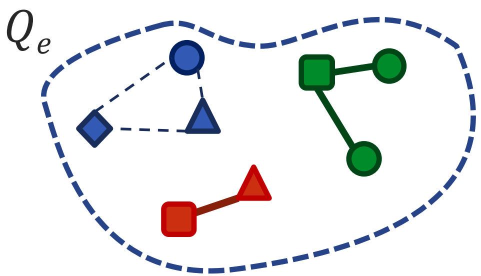
$$A : \{ a \in D \mid a \approx S \}$$
$\approx \rightarrow$ similarity

dbTrento

MULTIPLE EXAMPLE

Multi-Exemplar Queries

Our Problem Formulation:



Input: Q_e, a ***set of*** example ***elements*** of interest

Output: set of elements in the ***desired result set***

Exemplar Query Evaluation

- **match each** $q \in Q_e$ to the ***set of samples*** **S:{ s_1, s_2, …}** in the graph G
- **find** the set of ***elements A*** similar to ***each element in*** *S* given a ***similarity relation***

- [OPTIONAL] return only the ***top-k subset*** A^K

$$A : \{ a \in D \mid \forall s \in S \,.\, s \approx a \}$$

$\approx \rightarrow$ similarity

dbTrento

(Multi-)Exemplar Queries **on Graphs**

Single Sample $\mathbf{A : \{ \mathit{a} \in D \mid \mathit{a} \approx S \}}$

- **Similarity (≈) : graph isomorphism**

  A: { *a* <u>*subgraph of*</u> G | s <u>*isomorphic to*</u> *a*}

- **Challenge : find ALL isomorphic graphs**

<u>Graph Isomorphism is</u>
Transitive and Symmetric!
A is an Equivalence Class

Multiple Samples $\mathbf{A : \{ \mathit{a} \in D \mid \forall s \in S \,.\, s \approx \mathit{a} \}}$ $\mathbf{s_i, s_j \in S \,.\, s_i \neq s_j \rightarrow A = \varnothing}$

- **Similarity (≈) : ?Subgraph-Isomorphism**

  A: { *a* <u>*subgraph of*</u> G | ∀ s ∈ S . s <u>*subgraph isomorphic to*</u> *a*}

IS THIS CHARACTERIZATION ENOUGH?

Answers Are Subgraphs
That Contain Structures Similar To Each Sample

Multi-Exemplar **Answers on Graphs**

Graph Similarity (≈) **Subgraph-Isomorphism**

$\{ a \in G \mid \forall s \in S \,.\, s \;\underline{\textit{subgraph isomorphic to}}\; a \}$

What constitutes a good answer?

With No Restrictions the Entire Graph Is Accepted as Answer

dbTrento

Multi-Exemplar **Answers on Graphs**

Each answer should be
correct, **complete** and **non redundant**:

Ensure **all Aspects are present**
& **Limit Size** of Answer Graphs

Answers:
WEAKLY CONNECTED SUBGRAPHS
with **NO SUPERFLOUS NODES** or **EDGES**

1. Connectedness

$\forall \mathbf{n_1,n_2} \in \mathbf{V_A}\ \ \exists$ undirected path
that **connects** n_1 to n_2

✓ CORRECT

2. Consistency

$\forall n_A \in V_A \rightarrow \exists s \in S,\ n_s \in V_s$
Such that $\mathbf{n_A}$ **maps to** $\mathbf{n_s}$

✓ COMPLETE ✓ NON-REDUNDANT

dbTrento

CHALLENGE!

To find **Multi-Exemplar answers**

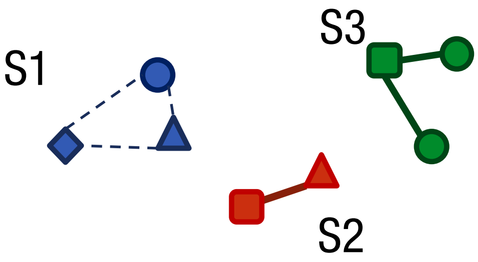
1. Find **ALL isomorphic graphs** to ALL samples
2. Find which samples **combine into one connected answer**

For each sample
needs to perform **Subgraph-Isomorphism Search**

Candidate space = Cartesian Product
for all samples

db Trento

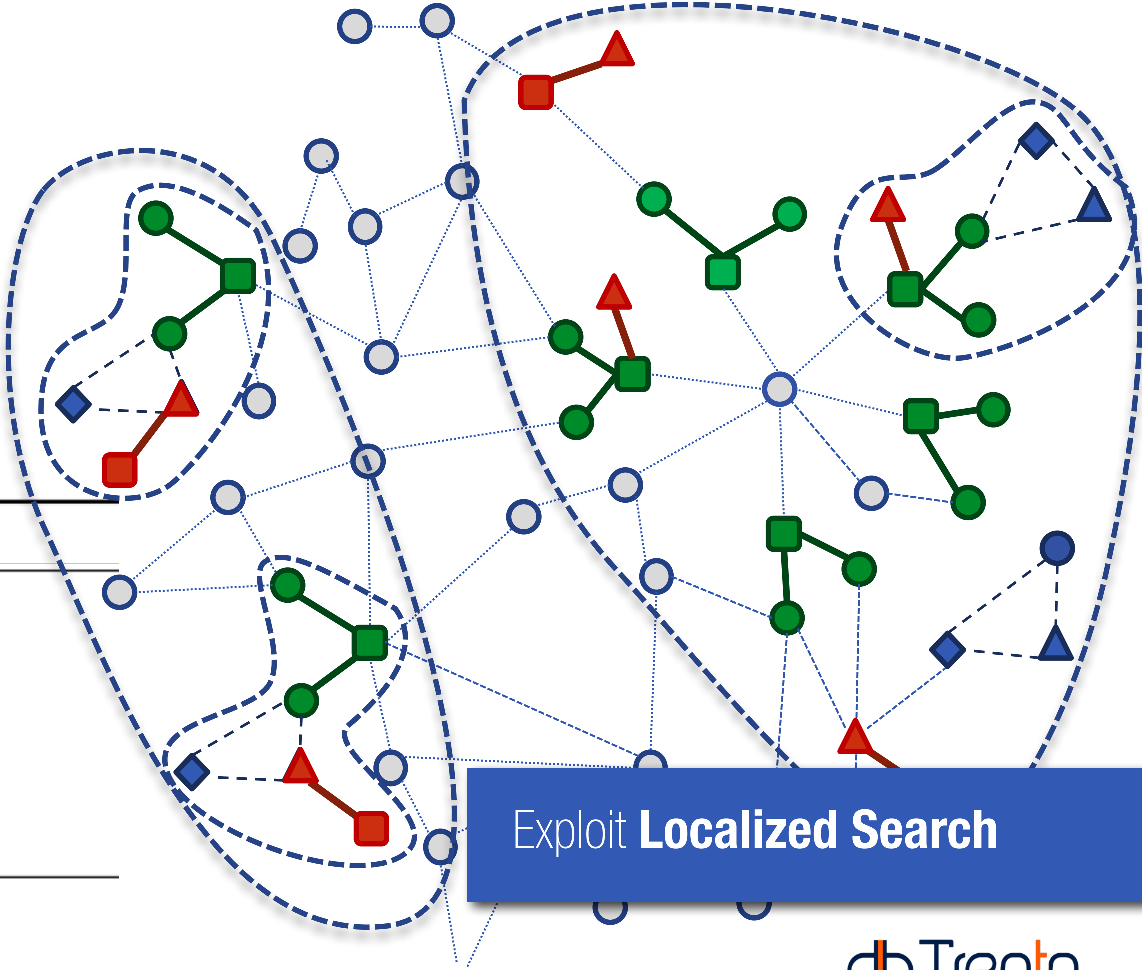
Search Framework



Multi-exemplar Answering

Input: Database $G : \langle V, E, \ell \rangle$
Input: Samples $\mathcal{S} : \langle s_1, \ldots, s_m \rangle$
Output: Answers $\mathcal{A}$

1: $\mathcal{G} \leftarrow \textsc{Partial}(G, \mathcal{S})$
2: $\mathcal{A} \leftarrow \textsc{Search}(\mathcal{G}, \mathcal{S})$
3: **return** $\mathcal{A}$



Exploit **Localized Search**

Search Framework **Optimizations**

1. **Find CANDIDATE REGIONS**
   a. **Remove Unused Edges**
   b. **Identify SEEDs**
   c. **Expand Around each seed**

2. **SEARCH within each region**
   a. **Avoid Cartesian Product**
   b. **Fast Merge of Partial Aswers**

Multi-exemplar Answering

Input: Database $G : \langle V, E, \ell \rangle$
Input: Samples $\mathcal{S} : \langle s_1, \ldots, s_m \rangle$
Output: Answers $\mathcal{A}$
1: $\mathcal{G} \leftarrow \textsc{Partial}(G, \mathcal{S})$
2: $\mathcal{A} \leftarrow \textsc{Search}(\mathcal{G}, \mathcal{S})$
3: **return** $\mathcal{A}$

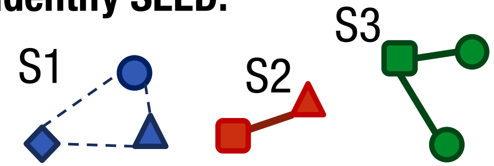
Naïve Algorithm
- **1 single** region
- Retrieves **ALL** Isomorphic-subgraphs
- **Hash-JOIN** for Fast merge of Partial Answers

2 Advanced Algorithms
- Fast & Fast+

dbTrento

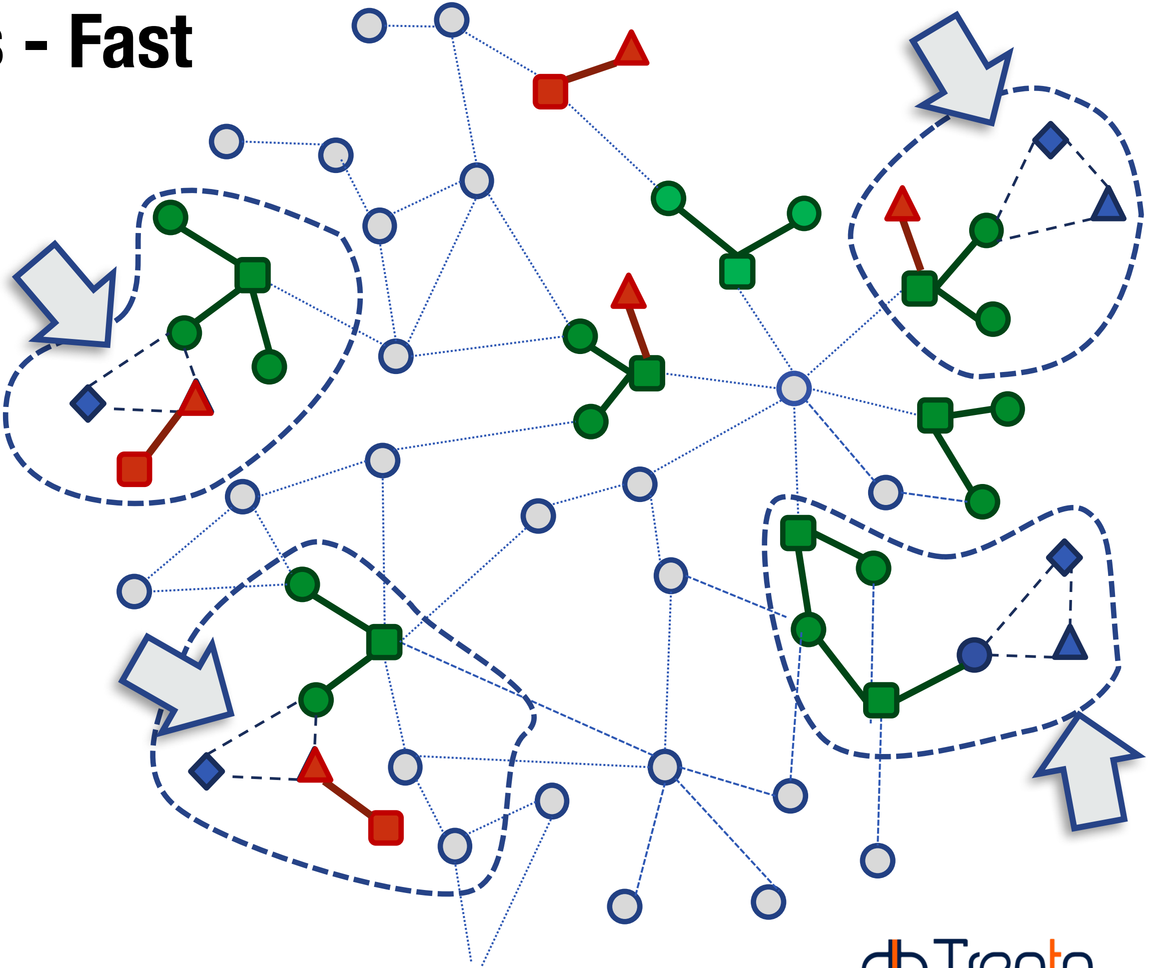
Find Candidate Regions - Fast

Identify SEED:



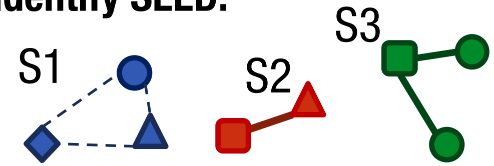
Min # of matches $\boxed{\text{S1: } \underline{\mathbf{4}}}$ S2: **6** S3: **7**

EXPAND around each seed:
Retrieve candidate Regions

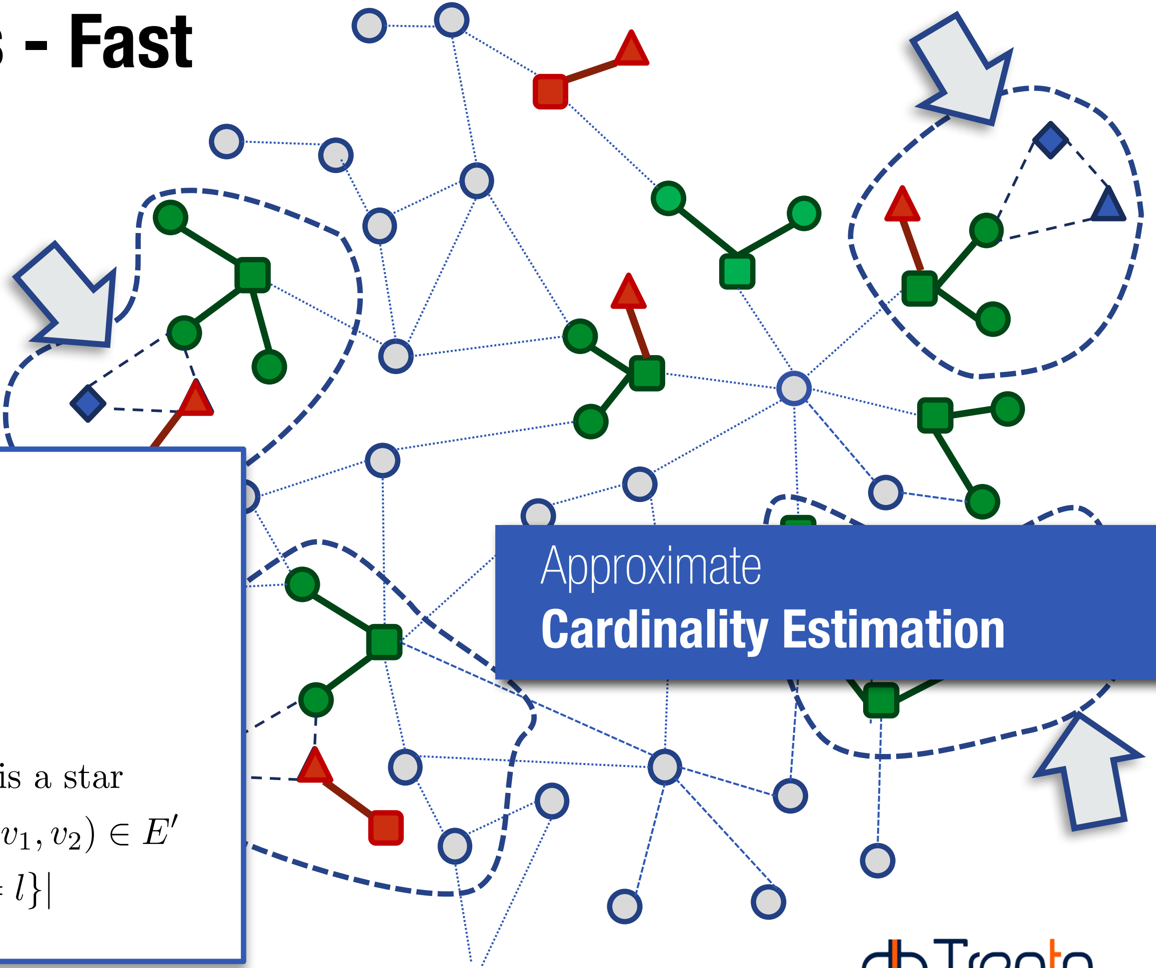


Find Candidate Regions - Fast

Identify SEED:



Min # of matches **S1: 4** S2: **6** S3: **7**



Precomputed Statistics:

- **Label frequency**
- **Label Pair frequency**
- **Star cardinality**

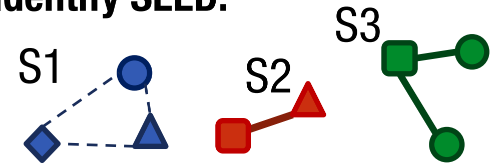
$$\mathbb{I}_{star}(l, c) = |\{G' \subseteq G | G' : \langle V', E', \ell \rangle \text{ is a star} \wedge |E'| = c \wedge \exists (v_1, v_2) \in E' \text{ s.t. } \ell(v_1, v_2) = l\}|$$

Approximate
Cardinality Estimation

dbTrento

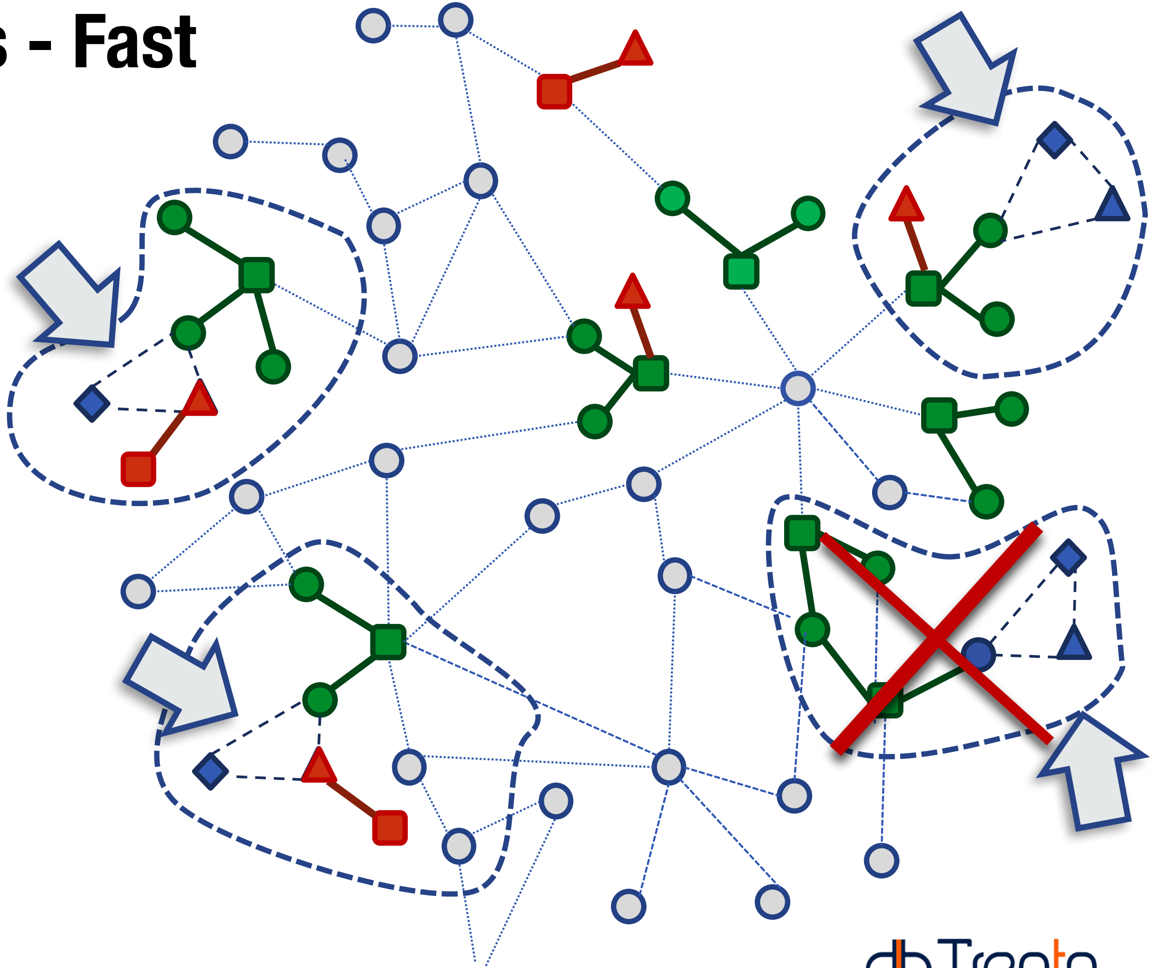
Find Candidate Regions - Fast

Identify SEED:



Min # of matches $\boxed{\text{S1: } \underline{\mathbf{4}}}$ S2: **6** S3: **7**

EXPAND around each seed:
Retrieve candidate Regions



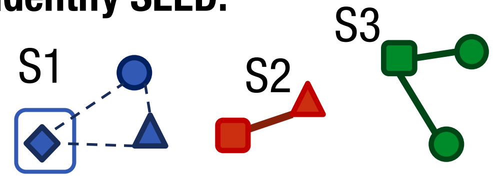
Seed search
requires Isomorphic-Search

Some Regions
do **not contain all structures**

dbTrento

Find Candidate Regions – Fast+

Identify SEED:

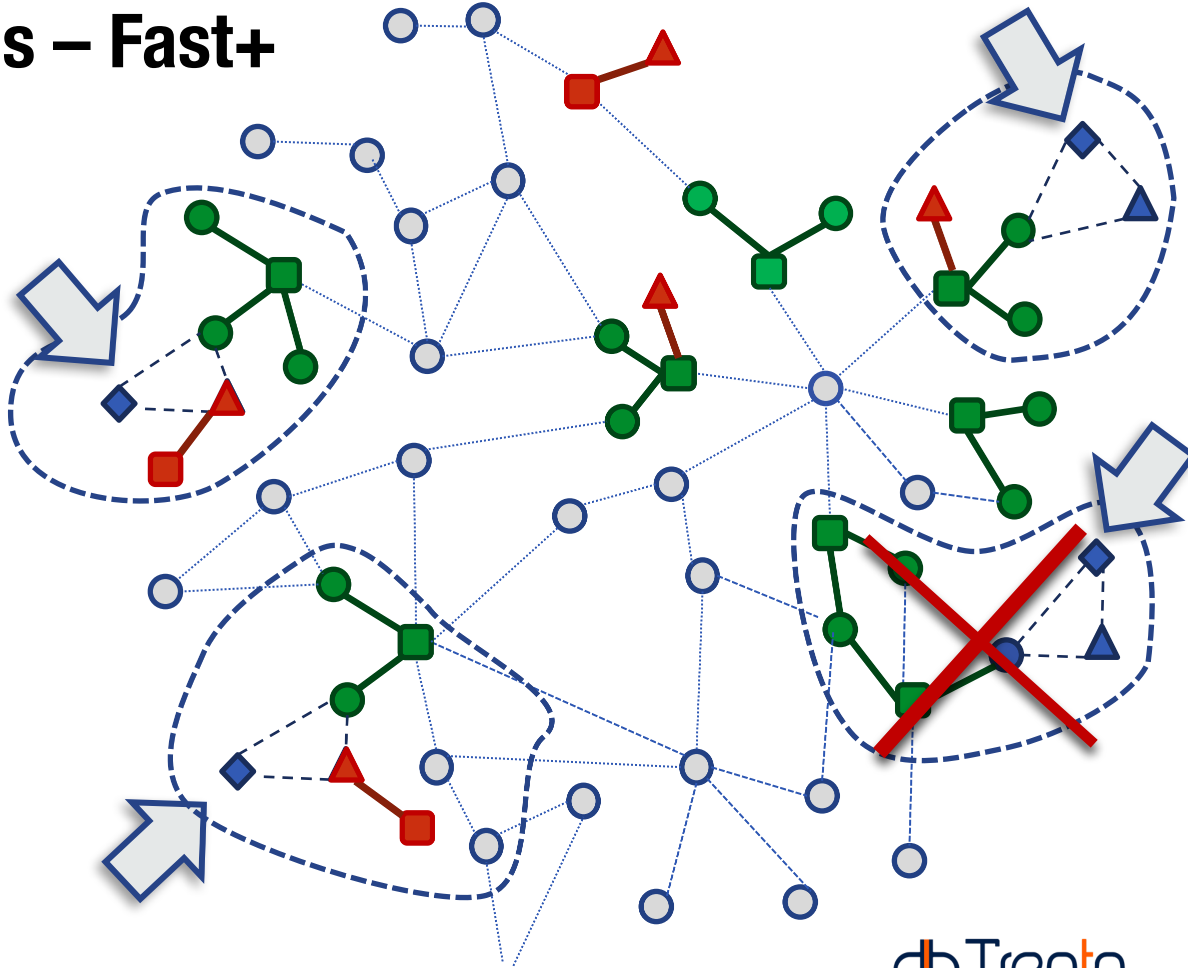


With cardinality Estimation

Select SINGLE NODE
With bitset-mapping

EXPAND around each seed:
Retrieve candidate Regions

DISCARD incomplete regions
With bitset-mapping & before ISO-search



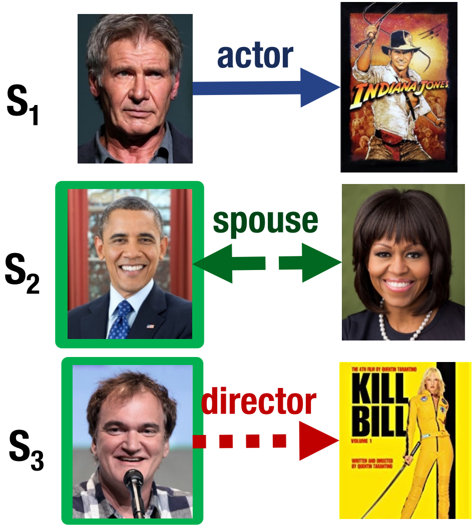
Fast Pruning with Bit-Vectors

$\overbrace{\text{actor} \quad \text{director} \quad \text{spouse} \quad \text{author}}^{d=1}$ . . .

	actor	director	spouse	author	
Barack Obama (S2)	-	-	1	-	
Quentin Tarantino (S3)	-	1	-	-	
Steven Spielberg (A1)	-	1	1	-	
Kate Capshaw (A2)	-	-	1	-	
BO ∨ QT	-	1	1	-	(union)
(BO ∨ QT) ∧ $\overline{\text{KC}}$	-	1	-	-	(≠0)
(BO ∨ QT) ∧ $\overline{\text{SS}}$	-	-	-	-	(=0)

Example:

Detect JOIN-Node



Top-K

Weight Function for Nodes

Scoring Function $$\rho(A) = \frac{1}{|V_A|} \sum_{v \in V_A} w(v)$$

Multi-exemplar Answering

Input: Database $G : \langle V, E, \ell \rangle$
Input: Samples $\mathcal{S} : \langle s_1, \ldots, s_m \rangle$
Output: Answers $\mathcal{A}$
1: $\mathcal{G} \leftarrow \textsc{Partial}(G, \mathcal{S})$
2: $\mathcal{A} \leftarrow \textsc{Search}(\mathcal{G}, \mathcal{S})$
3: **return** $\mathcal{A}$

Theorem 2. *Given the set of graph samples $\mathcal{S}$, and answers, A_1 and A_2, $\forall s \in \mathcal{S}.s \sqsubseteq A_i$, via the isomorphism function $\mu^s_{A_i}$, the node weight function w, and the ranking ρ (Equation 1). It holds:*

$$\overline{\rho}(A_2) = \frac{\sum_s^{\mathcal{S}} \sum_v^{V_s} w(\mu^s_{A_2}(v))}{max_{s \in \mathcal{S}} |V_s|} < \rho(A_1) \rightarrow \rho(A_2) < \rho(A_1) \quad (4)$$

Skip Regions that will not produce answers with score High Enough

dbTrento

Top-K

Weight Function for Nodes

Scoring Function $\rho(A) = \frac{1}{|V_A|} \sum_{v \in V_A} w(v)$

Multi-exemplar Answering

Input: Database $G : \langle V, E, \ell \rangle$
Input: Samples $\mathcal{S} : \langle s_1, ..., s_m \rangle$
Output: Answers $\mathcal{A}$
1: $\mathcal{G} \leftarrow$ PARTIAL$(G, \mathcal{S})$
2: $\mathcal{A} \leftarrow$ SEARCH$(\mathcal{G}, \mathcal{S})$
3: **return** $\mathcal{A}$

Theorem 2. *Given the set of graph samples $\mathcal{S}$, and answers, A_1 and A_2, $\forall s \in \mathcal{S}. s \sqsubseteq A_i$, via the isomorphism function $\mu^s_{A_i}$, the node weight function w, and the ranking ρ (Equation 1). It holds:*

$$\overline{\rho}(A_2) = \frac{\sum_s^{\mathcal{S}} \sum_v^{V_s} w(\mu^s_{A_2}(v))}{max_{s \in \mathcal{S}} |V_s|} < \rho(A_1) \rightarrow \rho(A_2) <$$

1. Sort regions (output of Partial)
2. Estimate **Upper Bound Score** for each
3. Find Top-K & prune

Skip Regions that will not produce answers with score High Enough

dbTrento

Experimental Evaluation

3 Algorithms:

a) **Naïve**
b) **Fast** (iso-graphs as seeds)
c) **Fast+** (nodes as seeds)

2 Large Real Datasets:

a) **YAGO +16.7M Edges**
b) **Freebase +300M Edges**

Tests: 100 Queries 2-5 Samples

- **Count isomorphic computations**
- **Running Time**

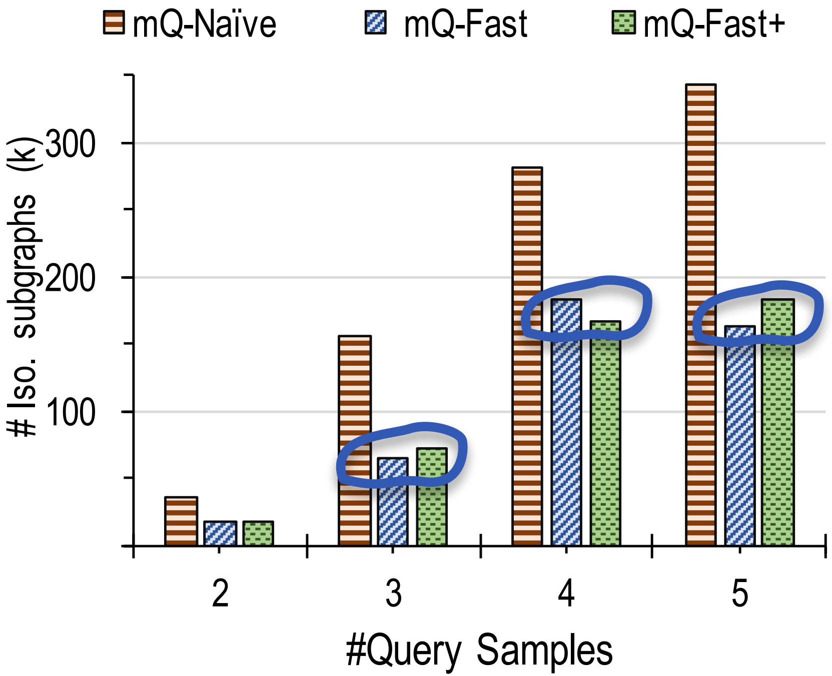
db Trento

Evaluation Results

Datasets: Freebase (300M Edges) Yago (16.7M edges)

100 queries, from 2 to 5 samples

Saving Isomorphic Computations



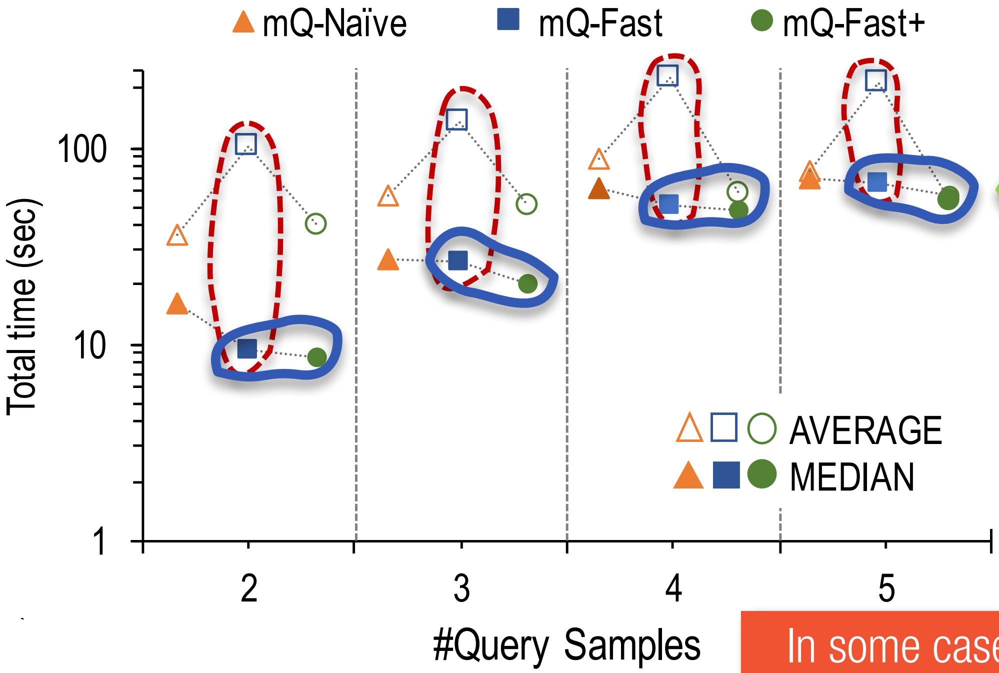
40-60% Reduction in Iso-Search Computations!

Evaluation Results

Datasets: Freebase (300M Edges) Yago (16.7M edges)

100 queries, from 2 to 5 samples

Running Times



Faster 70% of queries Saves up to 25secs on AVG

In some cases the "**Fast**" algorithm **wastes computations**

db Trento

Conclusions

Search via Multiple-Examples

Find **structures similar to a SET of input** examples.

Output: **Composite Results**
Containing characteristics from **each Example**

Useful When: a **Complete Example** with **all** the desired characteristics **is not known.**

Characteristics **combine** in **Multiple** ways.

Optimizations:

1. Exploit **Localized Search**
2. **Bitset Pruning**
3. **Cardinality Estimation**
4. **Top-K Optimizations**

Localized Search
Is less effective
With a Dense Graph!

bit.ly / icde18

Thank you!

Questions?

dbTrento

There is more . . .

Search Framework **Optimizations (bis)**

Algorithm 1 mQ-Naive

function PARTIAL$(G, \mathcal{S})$
  return $\mathcal{G} : \{G\}$ $\triangleright$ Selectively pruned based on $\mathcal{S}$
function SEARCH$(\mathcal{G}, \mathcal{S})$
  $\mathcal{A} \leftarrow \emptyset$
  for each $G \in \mathcal{G}$ **do**
    for each $s_i \in \mathcal{S}$ **do** $\triangleright$ Find isomorphic subgraphs
      $\tilde{A}_i \leftarrow \{A \subseteq G | s_i \approx A\}$
    $\mathbf{C} \leftarrow \arg\min_{\tilde{A}_i \in \langle \tilde{A}_1, \ldots, \tilde{A}_{|\mathcal{S}|} \rangle} |\tilde{A}_i|$
    while $\mathbf{C} \neq \emptyset$ **do**
      $c \leftarrow$ REMOVEONE$(\mathbf{C})$
      if $\forall s \in \mathcal{S}. s \sqsubseteq c$ **then** $\triangleright$ Check sample in c
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{c\}$
      else $\mathbf{C} \leftarrow \mathbf{C} \cup$ CONNECT$(c, \mathcal{S}, \langle \tilde{A}_1, \ldots, \tilde{A}_{|\mathcal{S}|} \rangle)$
  return $\mathcal{A}$

Algorithm 2 CONNECT$^+$

Input: Candidate $c : \langle V_c, E_c, \ell_c \rangle$; HashTable $\mathbb{H}$
Output: Expanded candidates C^+
$C^+ \leftarrow \emptyset$
$\triangleright$ Find candidate nodes contained in some answer
for each $n \in V_c$ **do**
  for each $\tilde{A}_i \in \mathbb{H}(n)$ s.t. $s_i \not\sqsubseteq c$ **do**
    $C^+ \leftarrow C^+ \cup$ MERGE$(c, \tilde{A}_i)$
return C^+

Multi-exemplar Answering

Input: Database $G : \langle V, E, \ell \rangle$
Input: Samples $\mathcal{S} : \langle s_1, \ldots, s_m \rangle$
Output: Answers $\mathcal{A}$
$\mathcal{G} \leftarrow$ PARTIAL$(G, \mathcal{S})$
$\mathcal{A} \leftarrow$ SEARCH$(\mathcal{G}, \mathcal{S})$
return $\mathcal{A}$

Naïve Algorithm

- **1 single** region
- Retrieves **ALL** Isomorphic-subgraphs
- **Hash-JOIN** for Fast merge of Partial Answers

2 Advanced Algorithms

- Fast & Fast+

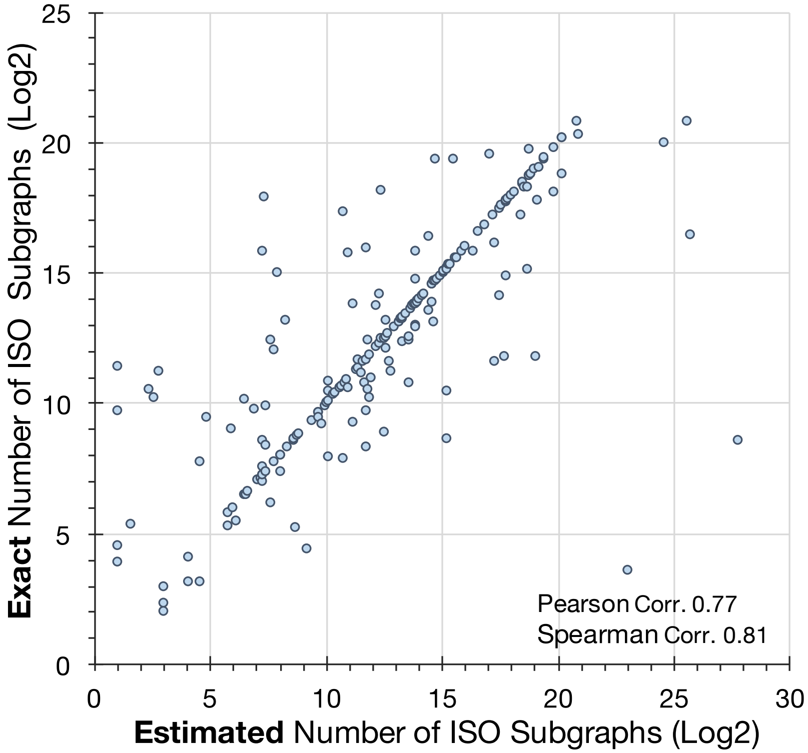
dbTrento

Cardinality Estimation

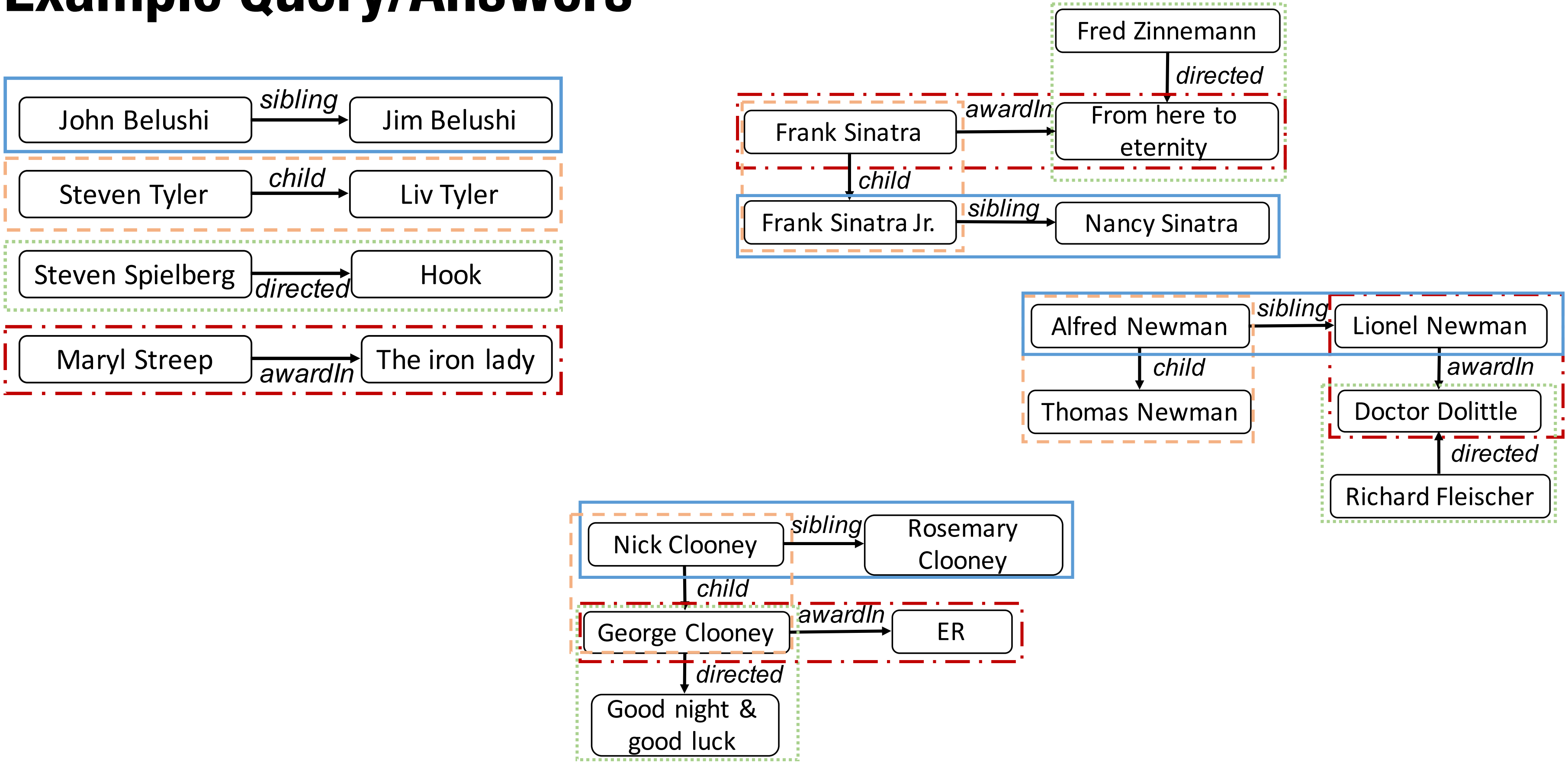
Precomputed Statistics:

- **Label frequency**
- **Label Pair frequency**
- **Star cardinality**

$$\mathbb{I}_{star}(l,c) = |\{G' \subseteq G | G' : \langle V', E', \ell \rangle \text{ is a star} \\ \wedge |E'| = c \wedge \exists (v_1, v_2) \in E' \\ \text{s.t. } \ell(v_1, v_2) = l\}|$$

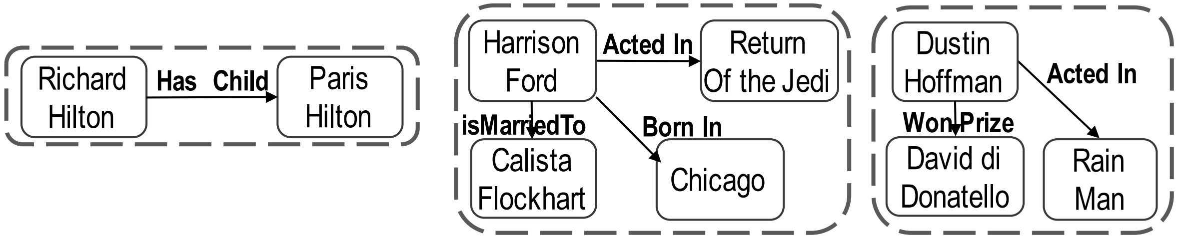


Example Query/Answers



Example Query/Answers

Examples



Answers