Review for “SANTOS: Relationship-based Semantic Table Union Search”

[SIGMOD’23]

Aamod Khatiwada*, Grace Fan*, Roee Shraga, Zixuan Chen,
Wolfgang Gatterbauer, Renée J. Miller, Mirek Riedewald
Northeastern University

Catalogue

CONTENTS

Introduction

Background ,Motivation, Challenges

Background

- **Data Lake**

  A centralized repository that stores large volumes of data in its natural or original form

- **Metadata**

  A description and definition of the data

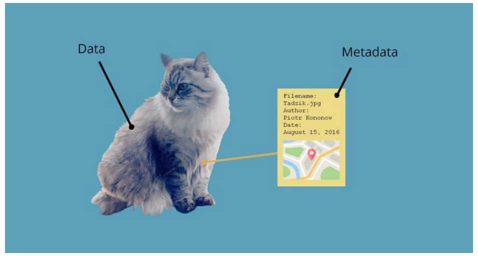
- **Knowledge Base**

  A broad and accurate knowledge graph that can model and represent entities, events, and relationships in the world

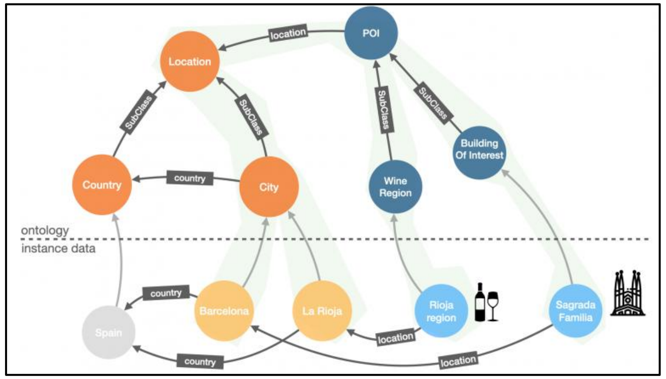
- **Table Union Search**

  Given a query table *Q* and a set of data lake tables *Ts*, the top-k table union search problem is to find the best k tables from *Ts* that can be unioned with the query table

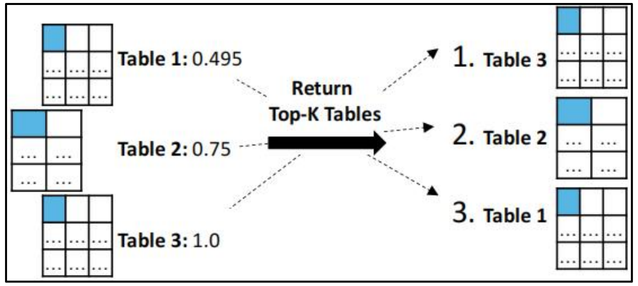
data lake



metadata



knowledge base



top-k table union search

Motivation

➢ Practical relevance

- Machine learning
- Data mining
- Information retrieval

➢ Dealing with data

- Metadata may be missing, inconsistent, or incomplete
- Deciding unionability based only on attribute unionability

Park Name	Supervisor	City	Country
River Park	Vera Onate	Fresno	USA
West Lawn Park	Paul Veliotis	Chicago	USA
--------	-------	--------	---------

(a)

Park Name	Film Title	Park Location	Park Phone	Park City	Film Director	Film Studio
Chippewa Park	Bee Movie	6748 N. Sacramento Ave.	773 731-0380	Cook	Simon J. Smith	Dreamworks
Lawler Park	Coco	5210 W. 64^{th} St.	773 284-7328	Riverside	Adrian Molina	Pixar
-----	------	--------	------	----	------	------

(b)

Person	Occupation	Birthplace	Country
James Taylor	Singer	Boston	USA
Anthony Pelissier	Film Director	Barnet	UK
Akram Afif	Football Player	Doha	Qatar
Ivan A. Getting	Physicist	NYC	USA
Abby May	Social Worker	Boston	USA
Stevie Ray Vaughan	Singer	Texas	USA

(c)

➢ Inspiration

- Using the data available within given tables instead of metadata to created the column and relationship semantics for table union search

Challenges & Solutions

➢ How to find column and relationship semantics accurately and comprehensively?

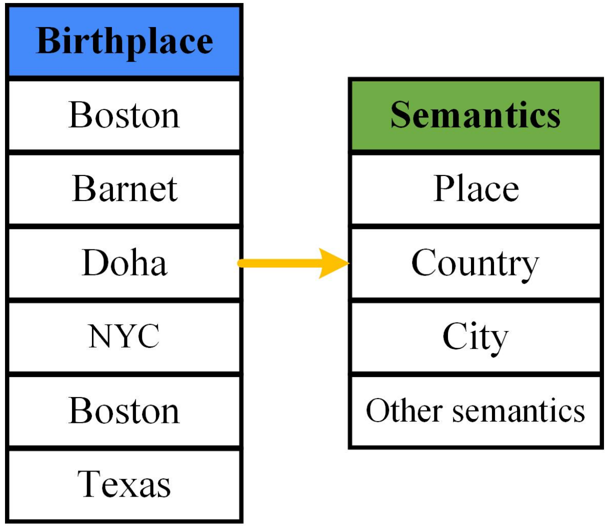
✓ Discover semantics with existing KB

✓ Discover semantics with synthesized KB

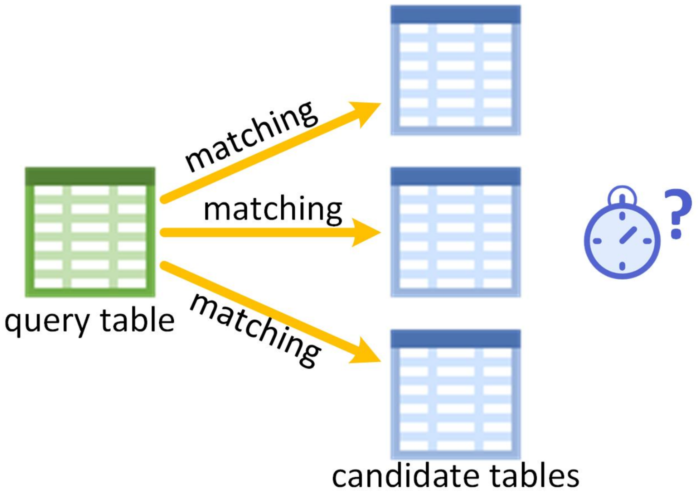
➢ How to quickly query the unionable tables while ensuring the search accuracy?

✓ present a scoring function to ensuring the search accuracy

✓ Semantics are created in the preprocessing phase, and only semantics are matched in the query phase



find semantics

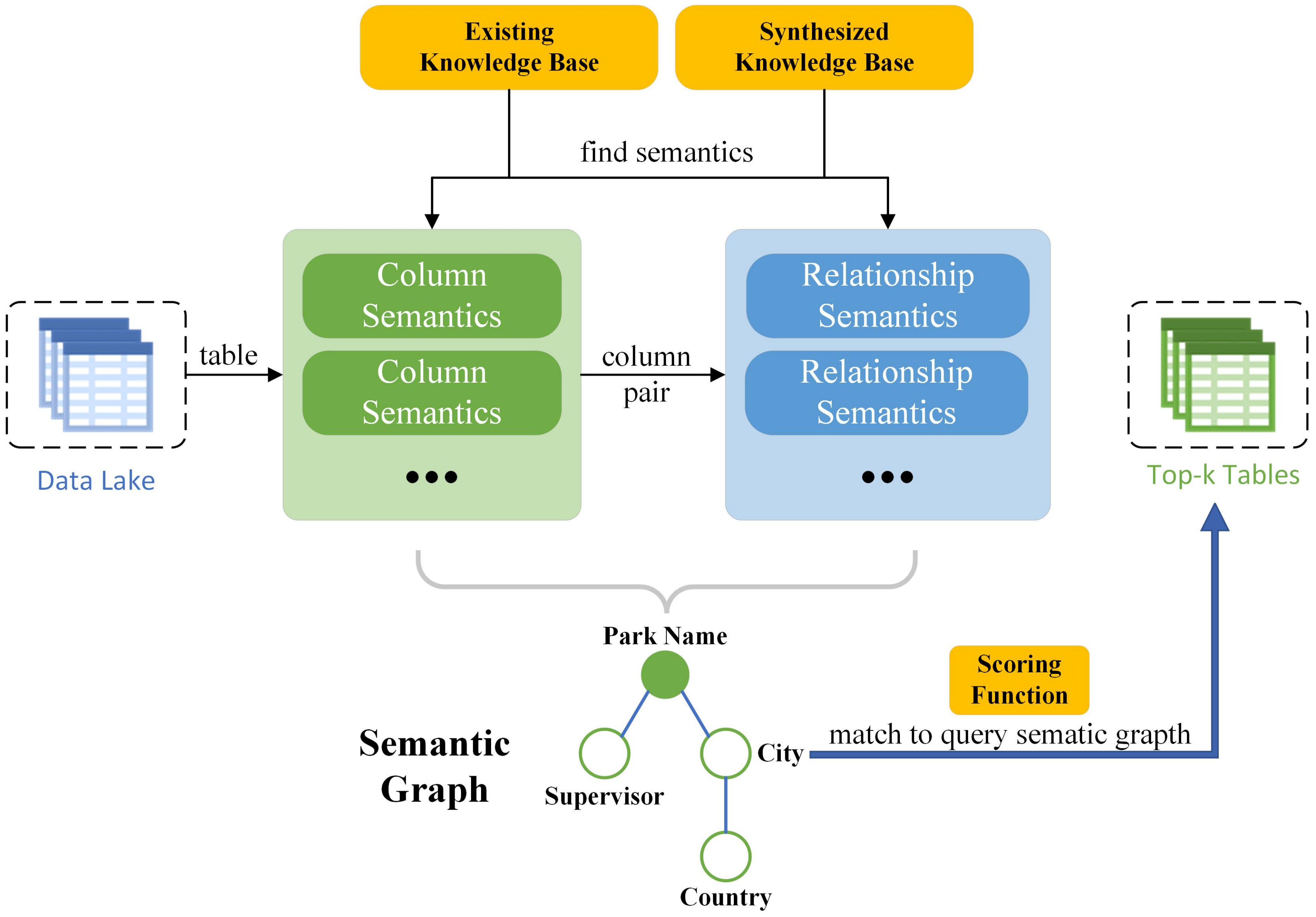


query phase

Methodology

Overview,Create Semantics，Semantics Graph,Union Search

Overview



Existing Knowledge Base
Synthesized Knowledge Base
find semantics
Column Semantics
Column Semantics
Relationship Semantics
Relationship Semantics
table
column pair
Data Lake
Top-k Tables
Park Name
Supervisor
City
Country
Semantic Graph
Scoring Function
match to query sematic grapth

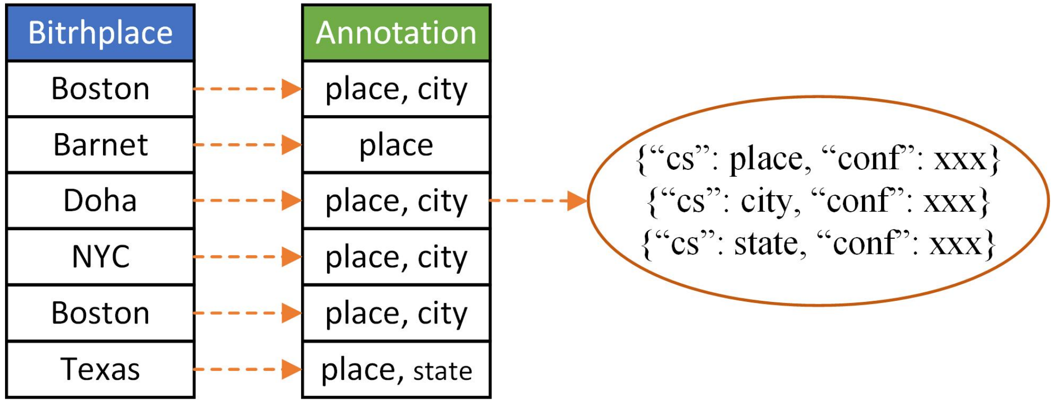
Create Semantics

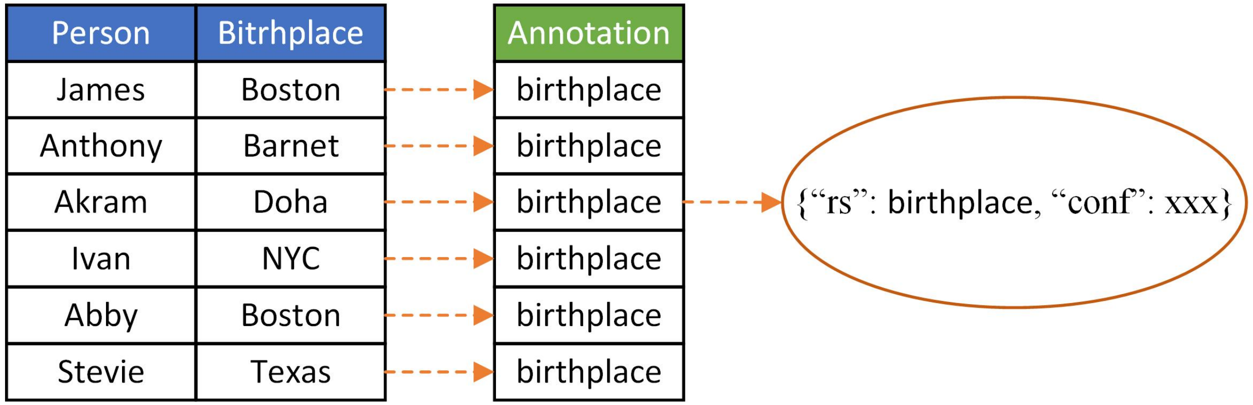
➢ Column Semantics [denoted *CS*(*c*)]

- Each column *c* in a table *T* has a set of semantic annotations
- Each annotation defines a conceptual domain to which the values in the column may belong
- Each annotation $a \in CS(c)$ has a confidence score

➢ Relationship Semantics [denoted $RS(c_1, c_2)$]

- Each pair of columns c_1, c_2 in a table *T* has a set of semantic annotations
- Each annotation defines a conceptual relationship to which the tuples in the pair of column may belong
- Each annotation a $\in RS(c_1, c_2)$ has a confidence score



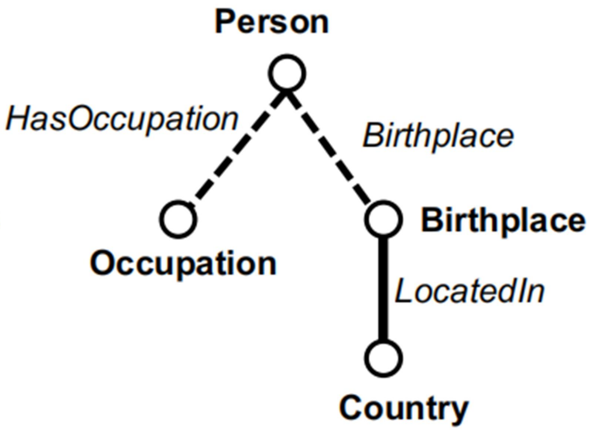


Semantic Graph

- **Semantic Graph**
- Nodes are the columns, for each column *c* labeled with *CS*(*c*)
- Edges are the relationships, for each relationship *r* labeled with *RS*(c_1, c_2)
- Edges connect pairs of columns if they have non-empty relationship semantics

Person	Occupation	Birthplace	Country
James Taylor	Singer	Boston	USA
Anthony Pelissier	Film Director	Barnet	UK
Akram Afif	Football Player	Doha	Qatar
Ivan A. Getting	Physicist	NYC	USA
Abby May	Social Worker	Boston	USA
Stevie Ray Vaughan	Singer	Texas	USA

(c)

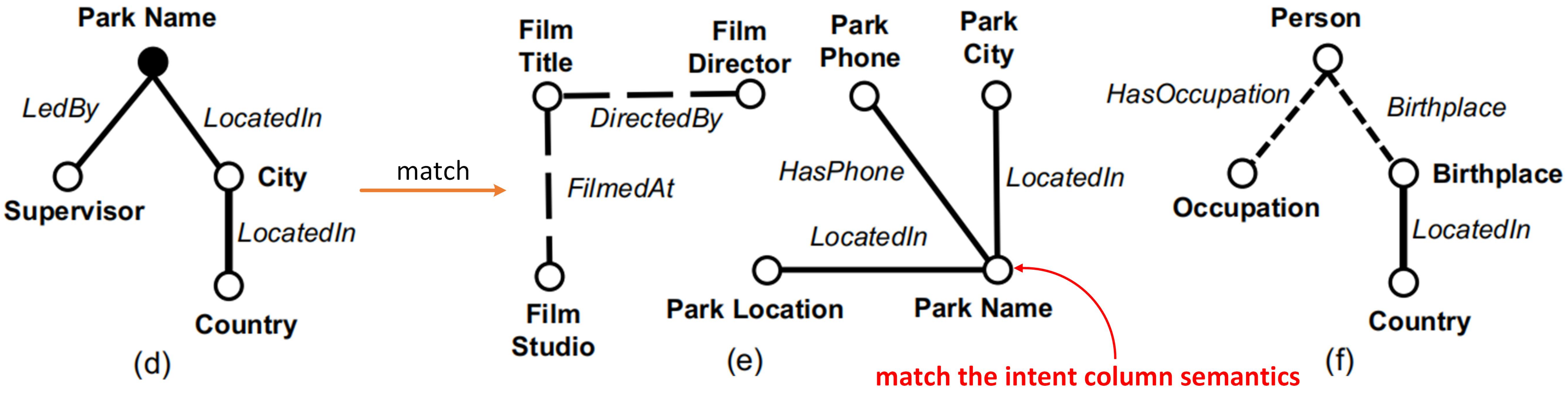


(f)

Semantic Graph

➢ Semantic Graph Matching

1. Given a query table and its intent column, forming a semantic graph that is restricted to being a tree rooted at the intent column
2. Looking for a tree within each semantic graph that matches a subtree of the query tree
3. Defining a scoring function that captures how closely the Semantic Graph of a data lake table matches with the Query Semantic Tree



Existing KB Semantic Graph

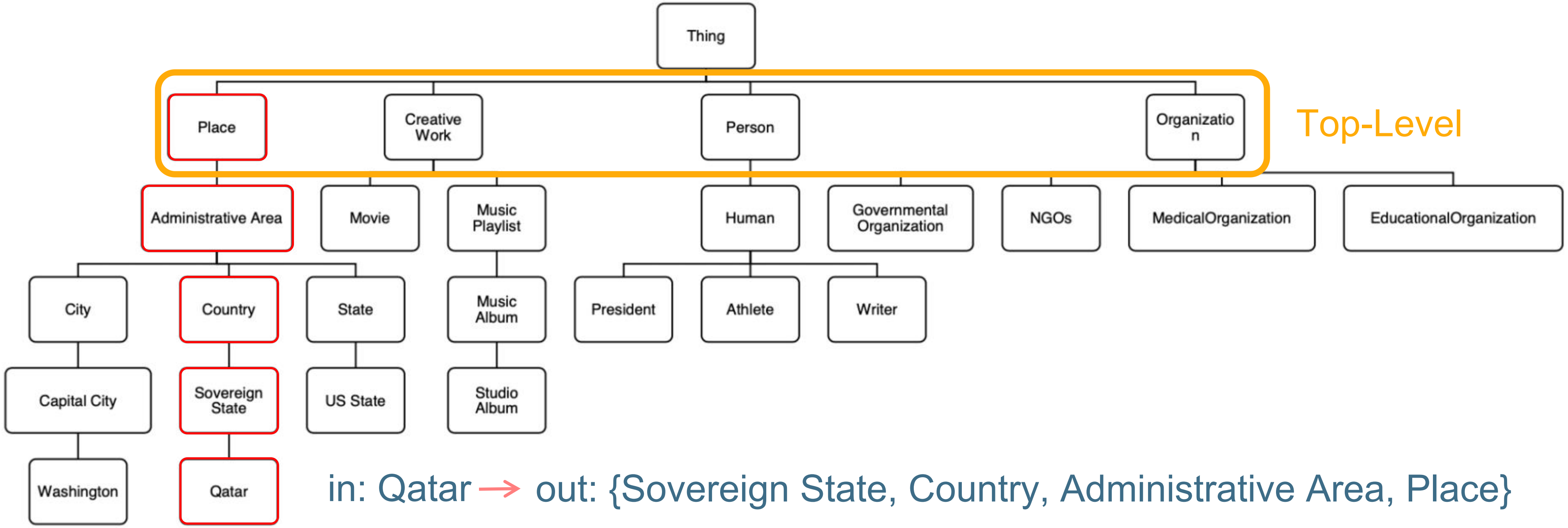
- ➢ **Knowledge Base**

Function: mapping an entity to semantic annotations

Intput: a cell value / a pair of value (metadata may be missing, inconsistent or incomplete)

Output: a set of annotations

Principle:CS is assigned based on semantic consistency



in: Qatar → out: {Sovereign State, Country, Administrative Area, Place}

Existing KB Semantic Graph

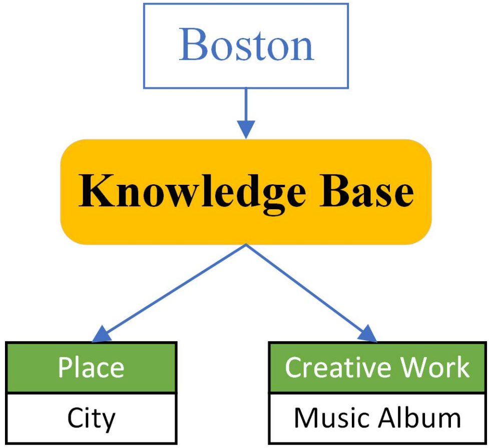
➢ **Knowledge Base**

Function: mapping an entity to an semantic annotation

Intput: a cell value / a pair of value

Output: a set of annotations

Principle:**CS is assigned based on semantic consistency**



Bitrhplace	Top-level
Boston	Place, Creative Work
Barnet	Place
Doha	Place
NYC	Place
Boston	Place, Creative Work
Texas	Place

the majority of the values are associated with **Place**
=> select **Place** and its descendants as the samantic annotions for the Birthplace column

Existing KB Semantic Graph

➢ Semantics Confidence

$$CS_{CONF}(c, a) = \begin{cases} \boxed{fs(a) \bullet gs(a)} & if\ c \in data-lake\ table\ T \\ fs(a) & if\ c \in query\ table\ Q \end{cases}$$

$$fs(a) = \frac{\boxed{|c_a|}}{\boxed{|c_{KB}|}}$$

$$gs(a) = \frac{1}{max(1, \boxed{log(a.count)})}$$

$$RS_{CONF}(c_i, p, c_j) = \frac{|(c_i, c_j)_p|}{|(c_i, c_j)_{KB}|}$$

Bitrhplace	Annotation
Boston	place, city
Barnet	place
Doha	place, city
NYC	place, city
Boston	place, city
Texas	place, state

$|c_{city}| = 3$

$|c_{KB}| = 5$

$fs(city) = 0.6$

city

Boston
Chicago
Houston
New York
…
Chicago

≈ 42,000 entities

$city.count \approx 42000$

$gs(city) \approx 0.22$

$CS_{CONF}(city) = 0.132$

$|c_a|$: The number of unique values mapped to annotation a from KB

$|c_{KB}|$: The total number of unique values mapped to the KB

a.count: The number of entities have annotation a

$|(c_i, c_j)_p|$: The number of unique value-pairs mapped to p from KB

$|(c_i, c_j)_{KB}|$: The total number of unique value-pairs mapped to the KB

Person	Bitrhplace	Annotation
James	Boston	birthplace
Anthony	Barnet	birthplace
Akram	Doha	birthplace
Ivan	NYC	birthplace
Abby	Boston	birthplace
Stevie	Texas	birthplace

$RS_{CONF}(birthplace) = 1.0$

Synthesized KB Semantic Graph

➢ Semantics Confidence

Limitaion of existing KB: KBs may have limited coverage over real data lakes. Hence, using only an existing KB (even a set of KBs) to determine CS and RS can lead to low coverage
Synthesized(Enhanced KB): Using the data lake itself, creating a synthesized KB

D
Kells Park
Eckhart Park
Union Park
Chopin Park
Wicker Park

- Depending on column D(may miss metadata) to create synthesized annotation named Annotation(D)
- CS(D) contains entities {Kells Park, Eckhart Park, Union Park, Chopin Park and Wicker Park}

$$CS_{CONF}(c,\ a \in CS(c_j)) = \begin{cases} 1 & if\ \ c = c_j \\ \dfrac{|c \cap c_j|}{|c|} & otherwise \end{cases}$$

$$RS_{CONF}(c_i, p, c_j) = \begin{cases} 1 & if\ \ c_i = c_j, d_i = d_j \\ \dfrac{|(c_i, c_j) \cap (d_i, d_j)|}{|(c_i, c_j)|} & otherwise \end{cases}$$

The number of unique values / pairs

Synthesized KB Semantic Graph

Table 1

A	B
Brands Park	***Moana***
Kells Park	***Spider-Man***
Eckhart Park	***Avengers***

RS(D, E): 0.66

Table 2

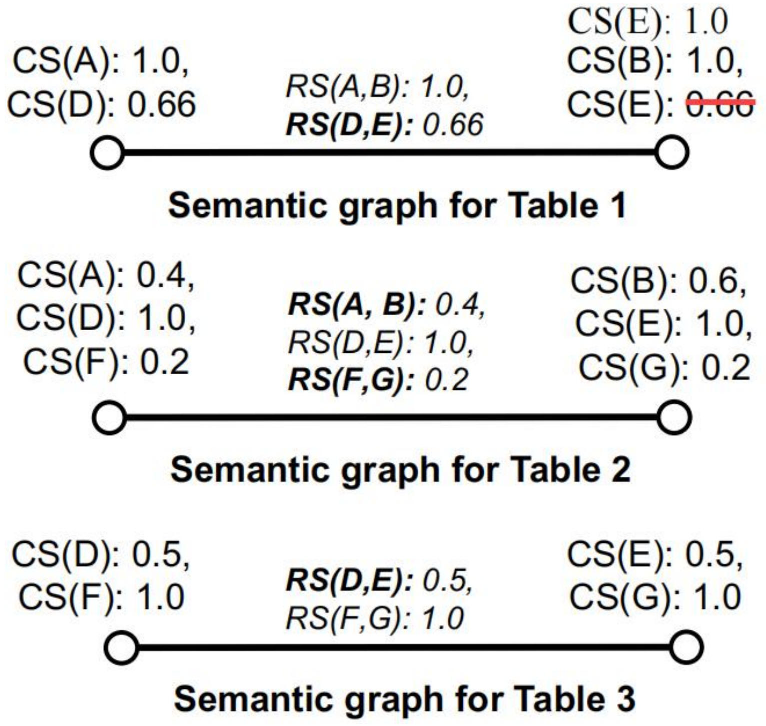
D	E
Kells Park	***Spider-Man***
Eckhart Park	***Avengers***
Union Park	*Black Panther*
Chopin Park	Trolls
Wicker Park	***Moana***

RS(A, B): 0.4 ***RS(F, G)***: 0.2

Table 3

F	G
Union Park	*Black Panther*
Gill Park	Wonder

RS(D, E): 0.5



$$CS_{CONF}(c,\ a \in CS(c_j)) = \begin{cases} 1 & if\ c = c_j \\ \dfrac{|c \cap c_j|}{|c|} & otherwise \end{cases}$$

$$RS_{CONF}(c_i,\ p, c_j) = \begin{cases} 1 & if\ c_i = c_j, d_i = d_j \\ \dfrac{|(c_i, c_j) \cap (d_i, d_j)|}{|(c_i, c_j)|} & otherwise \end{cases}$$

Let c = **Column D**, a = **Annotation A**, $|c| = 5, |c \cap c_j| = 2, CS_{CONF}(C_D, A_A \in CS(C_D)) = 0.4$

Let c = **Column D**, a = **Annotation D**, $c = c_j, CS_{CONF}(C_D, A_D \in CS(C_D)) = 1$

Let c = **Column D**, a = **Annotation F**, $|c| = 5, |c \cap c_j| = 1, CS_{CONF}(C_D, A_F \in CS(C_D)) = 0.2$

Synthesized KB Semantic Graph

Table 1

A	B
Brands Park	*Moana*
Kells Park	*Spider-Man*
Eckhart Park	*Avengers*

RS(D, E): 0.66

Table 2

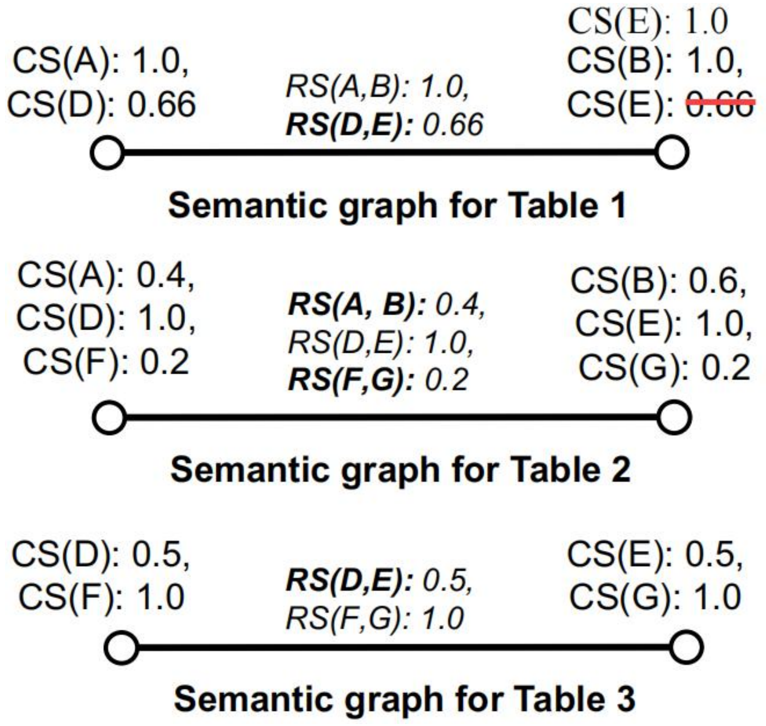
D	E
Kells Park	*Spider-Man*
Eckhart Park	*Avengers*
Union Park	*Black Panther*
Chopin Park	Trolls
Wicker Park	*Moana*

RS(A, B): 0.4 ***RS(F, G)***: 0.2

Table 3

F	G
Union Park	*Black Panther*
Gill Park	Wonder

RS(D, E): 0.5



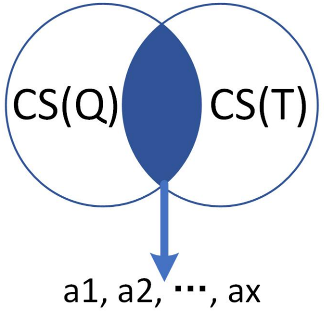
$$CS_{CONF}(c,\ a \in CS(c_j)) = \begin{cases} 1 & if\ c = c_j \\ \dfrac{|c \cap c_j|}{|c|} & otherwise \end{cases}$$

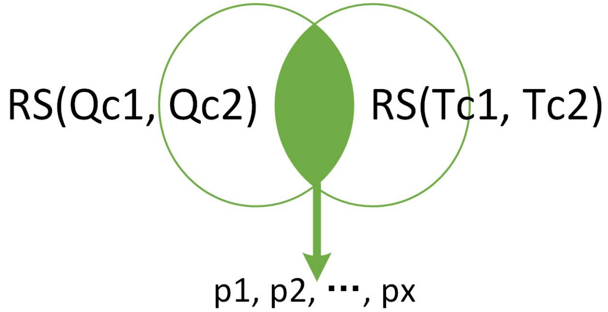
$$RS_{CONF}(c_i,\ p, c_j) = \begin{cases} 1 & if\ c_i = c_j, d_i = d_j \\ \dfrac{|(c_i, c_j) \cap (d_i, d_j)|}{|(c_i, c_j)|} & otherwise \end{cases}$$

Let c_i = **Column D**, c_j = **Column E**, p = **Annotation A-B**

$|(c_i, c_j)| = 5, |(c_i, c_j) \cap (d_i, d_j)| = 2, RS_{CONF}(C_D, A_{A-B}, C_E) = 0.4$

Union Search





Step-1 Filtering candidate table from data lake

Step-2 Calculating column & relationship & pair match confidence score

$$colMatch_G(Q_c, T_c) = \max_i CS_{CONF}(Q_c, a_i) \cdot CS_{CONF}(T_c, a_i)$$

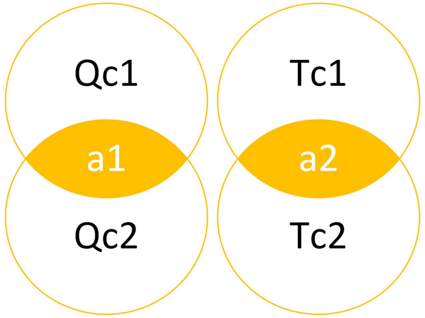
$$relMatch_G((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) = \max_i RS_{CONF}(Q_{c1}, p_i, Q_{c2}) \cdot RS_{CONF}(T_{c1}, p_i, T_{c2})$$

$$pairMatch_G((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) = colMatch_G(Q_{c1}, T_{c1}) \cdot relMatch_G((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) \cdot colMatch_G(Q_{c2}, T_{c2})$$

Step-3 Existing KB VS Synthesized KB (pair match confidence score comparison)

$$pairMatch((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) = \begin{cases} pairMatch_{KB}((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) & if\ f = 1 \\ pairMatch_{Synth}((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) & otherwise \end{cases}$$

$$f = 1\ if\ \frac{pairMatch_{KB}((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2}))}{gs(a_1) \cdot gs(a_2)} \geq pairMatch_{Synth}((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2}))$$



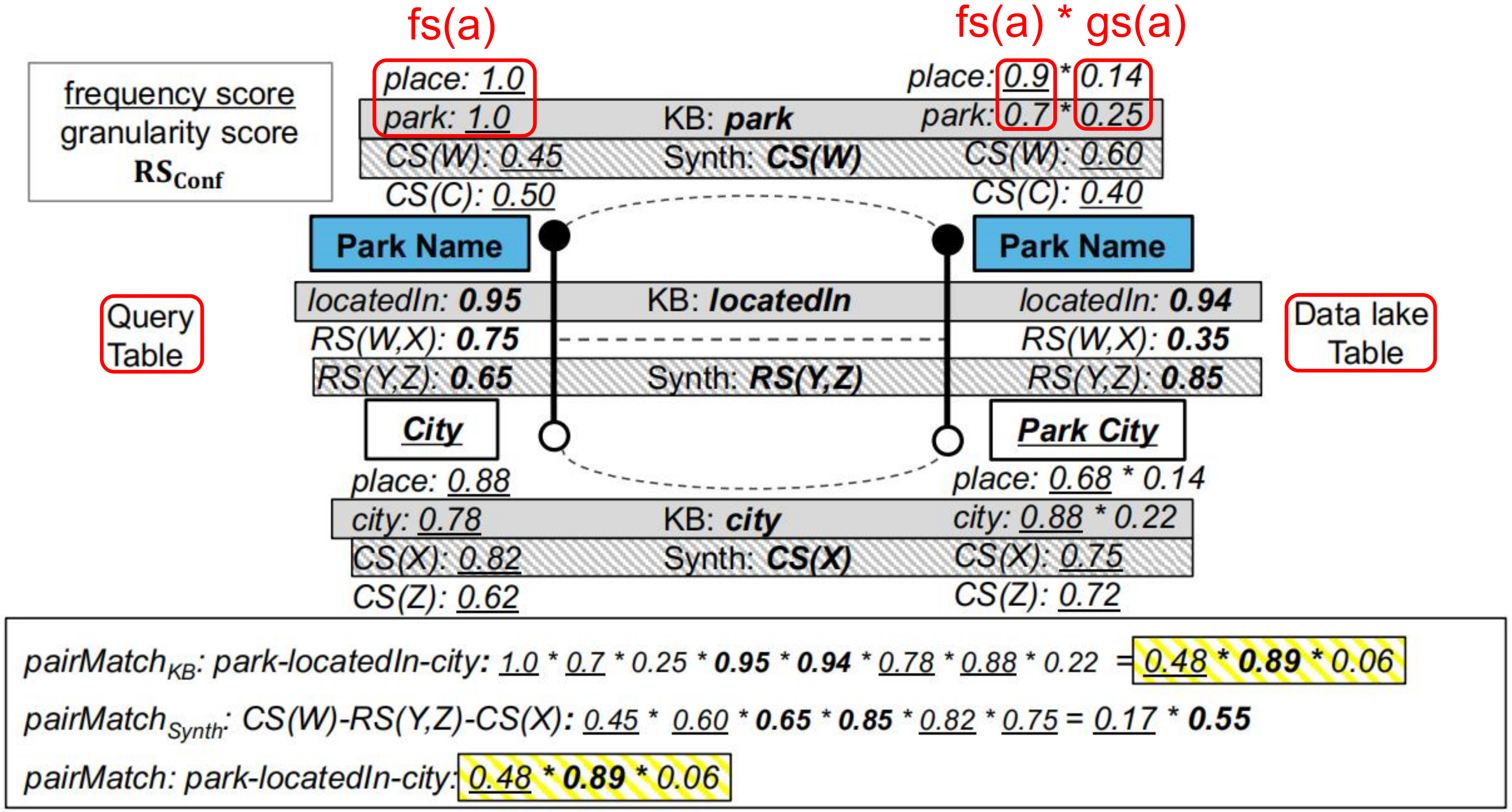
Step-4 Accumulating all pairs match confidence as table match confidence score

$$S(Q, T) = \sum_{i=1}^{m} pairMatch(Q.l, Q.c_i), (T.c, T.c_i))$$

Step-5 Returning tables that have top-k table match confidence score

Union Search

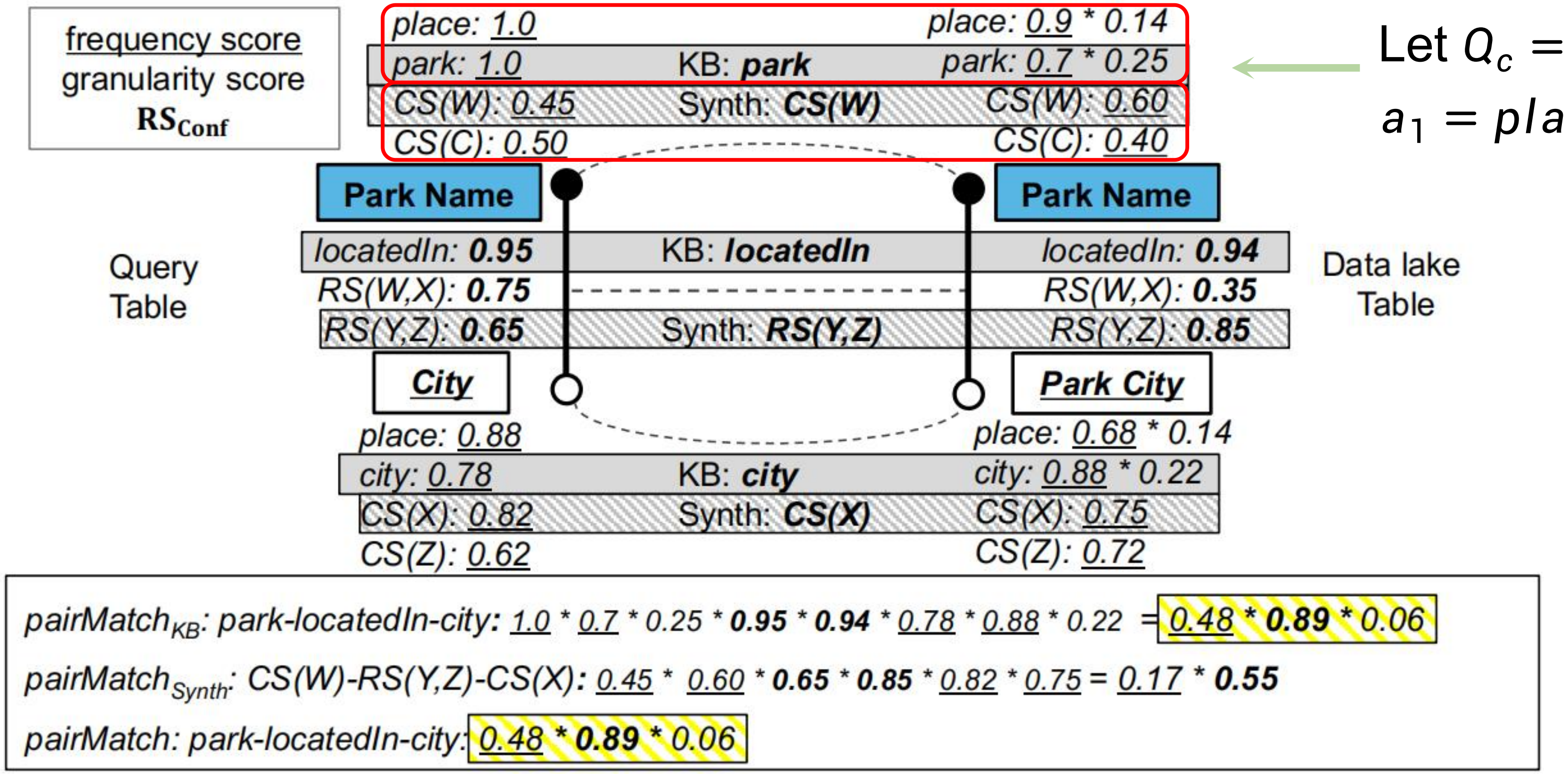
$$CS_{CONF}(c,a) = \begin{cases} \boxed{fs(a) \bullet gs(a)\ if\ c \in data-lake\ table\ T} \\ \boxed{fs(a) \qquad\qquad if\ c \in query\ table\ Q} \end{cases}$$



On the left is the query table; according to the formula, the confidence of the column semantic annotation is equivalent to the frequency score. On the right is the data lake table, where the confidence of the column semantic annotation is the product of the frequency score and the granularity score.

Union Search

$$colMatch_G(Q_c, T_c) = \max_i CS_{CONF}(Q_c, a_i) \cdot CS_{CONF}(T_c, a_i)$$



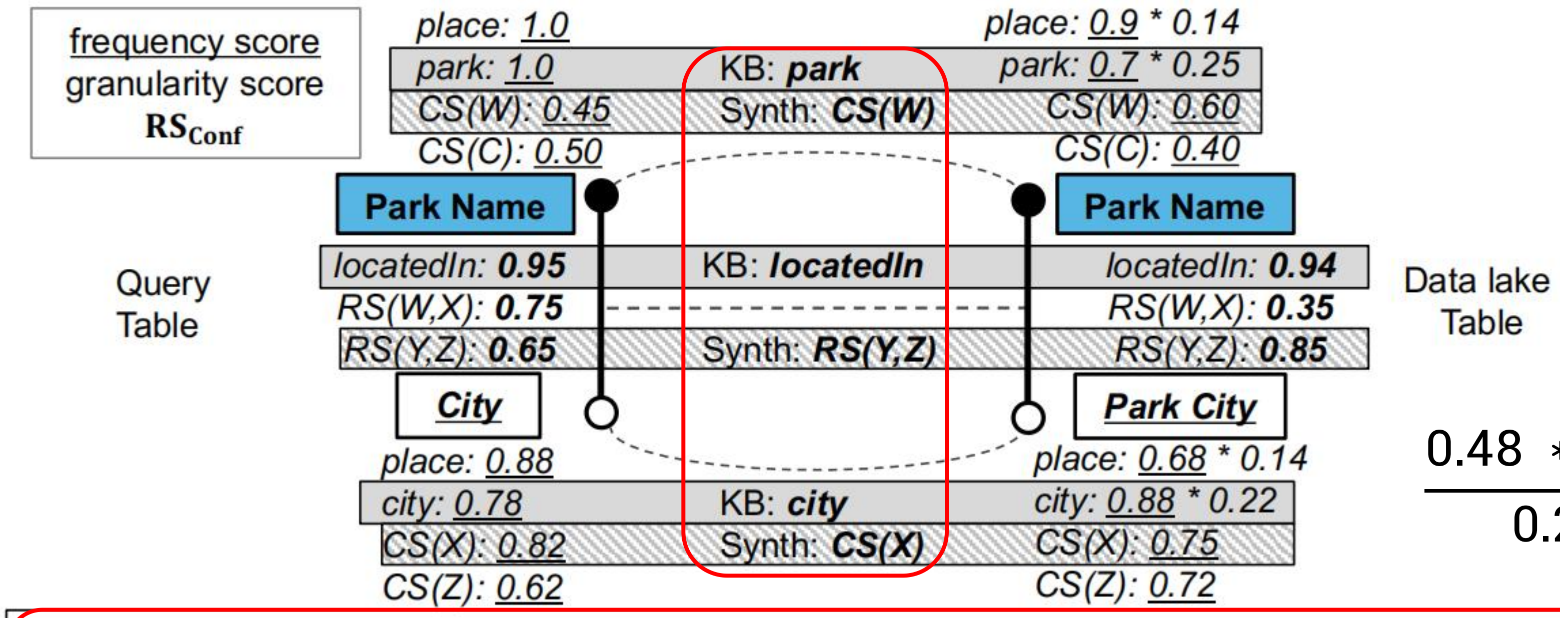
Let $Q_c = Q_{Park\ Name}$, $T_c = T_{Park\ Name}$

$a_1 = place$, $a_2 = park$, $a_3 = W$, $a_4 = C$

$$colMatch_{KB} = max(1.0 * 0.9 * 0.14, \quad 1.0 * 0.7 * 0.25)$$

$$colMatch_{Synth} = max(0.45 * 0.6, \quad 0.5 * 0.4)$$

Union Search



$$\frac{0.48 * 0.89 * 0.06}{0.25 * 0.22} > 0.17 * 0.55$$

$pairMatch_{KB}$: park-locatedIn-city: 1.0 * 0.7 * 0.25 * **0.95** * **0.94** * 0.78 * 0.88 * 0.22 = 0.48 * **0.89** * 0.06

$pairMatch_{Synth}$: CS(W)-RS(Y,Z)-CS(X): 0.45 * 0.60 * **0.65** * **0.85** * 0.82 * 0.75 = 0.17 * **0.55**

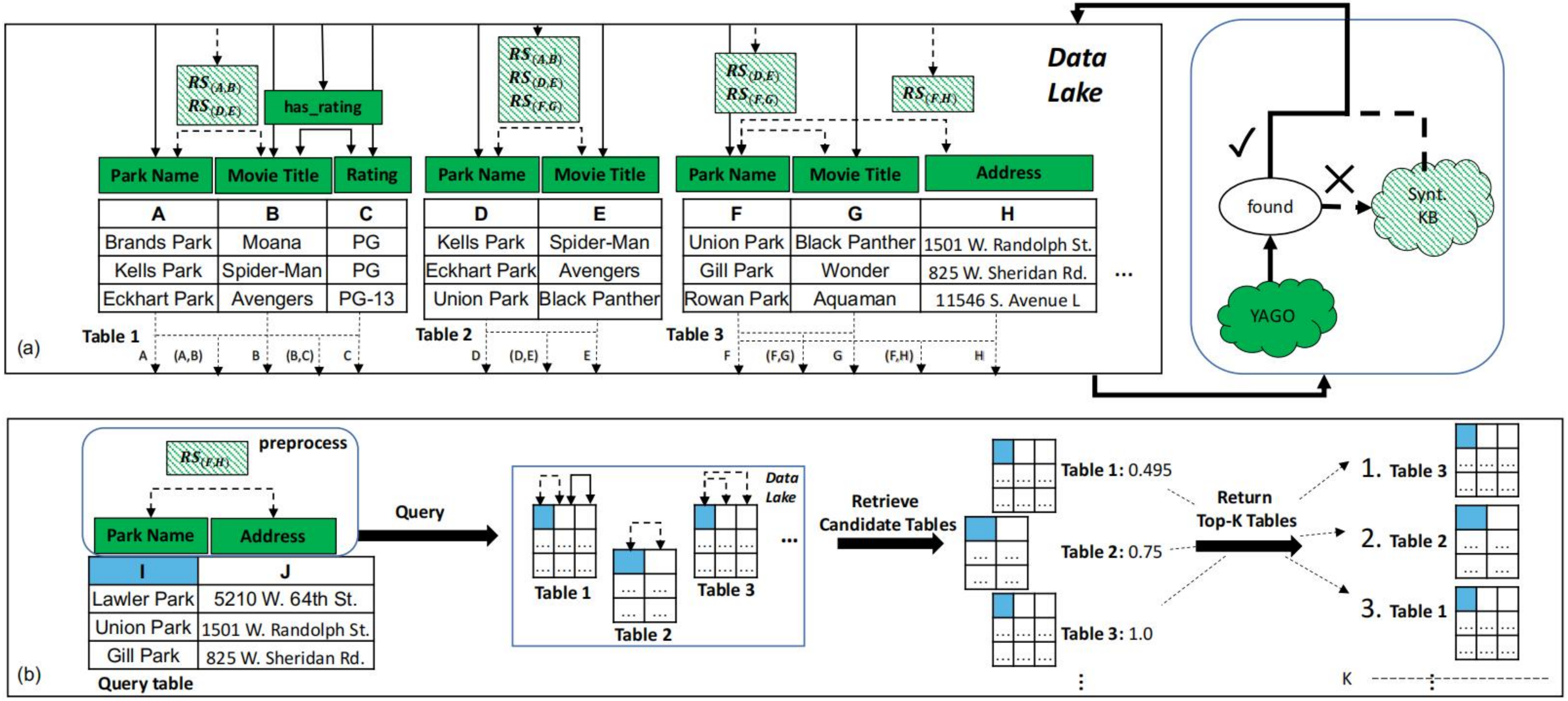
pairMatch: park-locatedIn-city: 0.48 * **0.89** * 0.06

$$pairMatch((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) = \begin{cases} pairMatch_{KB}((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) & if\ f = 1 \\ pairMatch_{Synth}((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2})) & otherwise \end{cases}$$

$$f = 1\ if\ \frac{pairMatch_{KB}((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2}))}{gs(a_1) \cdot gs(a_2)} \geq pairMatch_{Synth}((Q_{c1}, Q_{c2}), (T_{c1}, T_{c2}))$$

Pipeline of SANTOS

Preprocessing phase: data-lake tables are labeled with semantic annotations from KB
Query phase: the query table is annotated, and SANTOS queries the data lake to retrieve and rank unionable tables



Experiments

Comparison with Baselines

Experimental Setup

➢ Evaluation Measures

$$P@k = \frac{\mathcal{T}_Q \cap \hat{\mathcal{T}}_Q}{\hat{\mathcal{T}}_Q} \qquad R@k = \frac{\mathcal{T}_Q \cap \hat{\mathcal{T}}_Q}{\mathcal{T}_Q} \qquad MAP@k = \frac{1}{|\hat{\mathcal{T}}_Q|} \sum_{k=1}^{|\hat{\mathcal{T}}_Q|} P@k$$

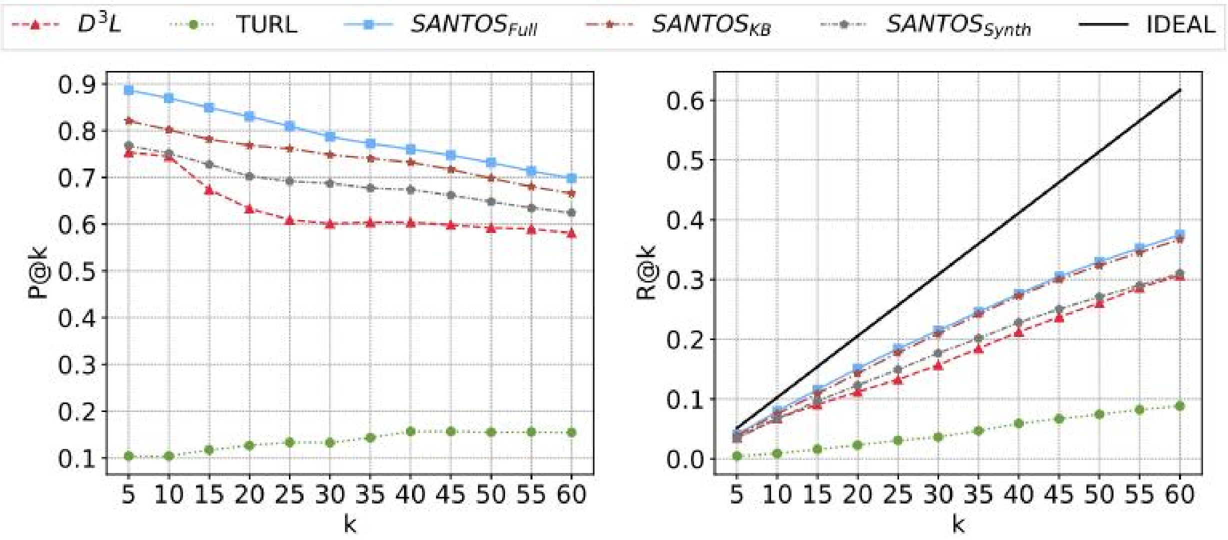
➢ Datasets

Table Source	Data lake Tables			Query Tables		
	# Tables	# Columns	# Rows	# Tables	# Columns	# Rows
TUS	1,530	14,810	6.8 M	125	1,610	557 K
SMALL	550	6,322	3.8 M	50	615	1.07 M
LARGE	11,090	123,477	70 M	80	1,017	1.03 M

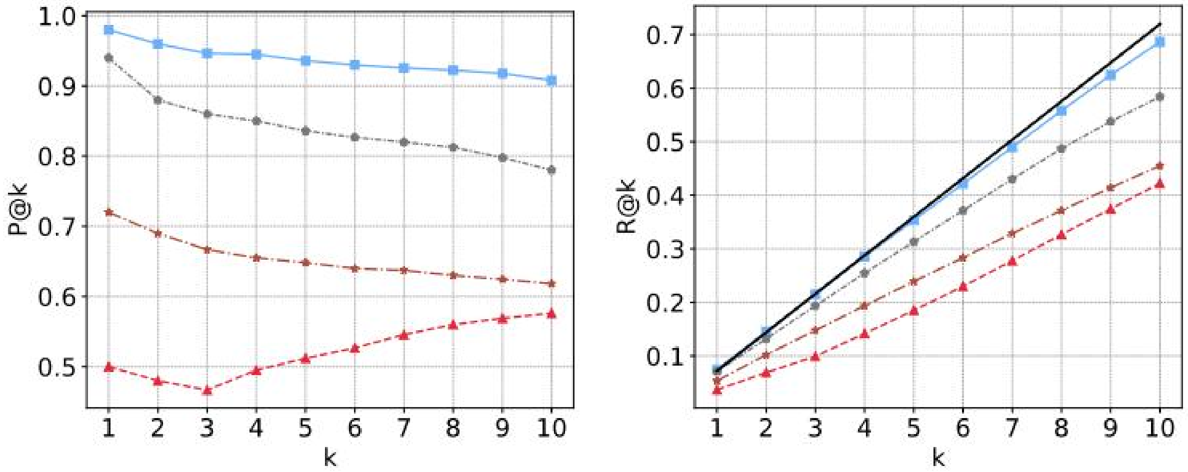
➢ Baselines

- D^3L adds metrics based on column names, regular expressions and domain distributions to the word-embedding and value overlap-based models
- TURL is a recent method that uses representational learning over web tables

Experimental Evaluation



(a) Average *P@k* on TUS (b) Average *R@k* on TUS



(c) Average *P@k* on SMALL (d) Average *R@k* on SMALL

Benchmark	Method	MAP@k	P@k	R@k
TUS (k=60)	TURL	0.13	0.16	0.08
	D^3L	0.64	0.58	0.31
	SANTOS	**0.80**	**0.70**	**0.37**
SMALL (k=10)	D^3L	0.52	0.58	0.42
	SANTOS	**0.93**	**0.90**	**0.68**
LARGE (k=20)	D^3L	0.29	0.26	-
	SANTOS	**0.77**	**0.73**	**-**

Summary: all indicators of SANTOS are better than baselines

Experimental Evaluation

Benchmark	Method	Indexing	Query (sec)
TUS	D^{3}L	1 hr 21 min	54.1 (20.5 – 97.3)
	SANTOS$_{Full}$	4 hr 26 min	22.9 (1.7 – 48.6)
	SANTOS$_{KB}$	1 hr 38 min	6.1 (0.7 – 13.9)
	SANTOS$_{Synth}$	3 hr 45 min	15.6 (0.7 – 43.2)
SMALL	D^{3}L	17 min	22.4 (7.4 – 43.3)
	SANTOS$_{Full}$	4 hr 46 min	28.2 (0.8 – 102)
	SANTOS$_{KB}$	1 hr 8 min	10.0 (0.3 – 33.6)
	SANTOS$_{Synth}$	3 hr 41 min	18.2 (0.5 – 98.6)
LARGE	D^{3}L	7 hr 7 min	177 (13.0 – 325.0)
	SANTOS$_{Full}$	21 hr 59 min	35.8 (0.21 – 57.2)

Summary: query time of SANTOS faster than the state-of-the-art approach while ensuring query accuracy

Conclusion

Conclusion & Limitation

Conclusion & Limitation

➢ Conclusion

- Relationship semantics is important in union search
- Effectiveness, the accuracy of top-k union search has been improved by SANTOS
- Scalability, suitable for data lakes of all scales

➢ Limitation

- The time overhead of indexing is high, which means that the table in the data lake can't be modified frequently
- SANTOS requires the user to provide a query table and an intent column, which is not as convenient as the keyword search method

Example
To query infomation about IT books from the data lake, for keyword search, you only need to input “it book”. However, for SANTOS, you would need to find or create a query table and add some IT book titles to the intent column, such as “Computer Networks”, “C Primer”, “Linux Manual” and so on.

THANK YOU

Group Number: Group 3

Report time：2023/11/25

Chenjie Li，Flora Zhang，Dingyi Zhao