Table Discovery and Integration in Data Lakes

Aamod Khatiwada

Fourth Year PhD Candidate

Northeastern University, Khoury College of Computer Sciences, Boston

Advised by: Prof. Renée J. Miller

March 18, 2024

Data, lots of data…

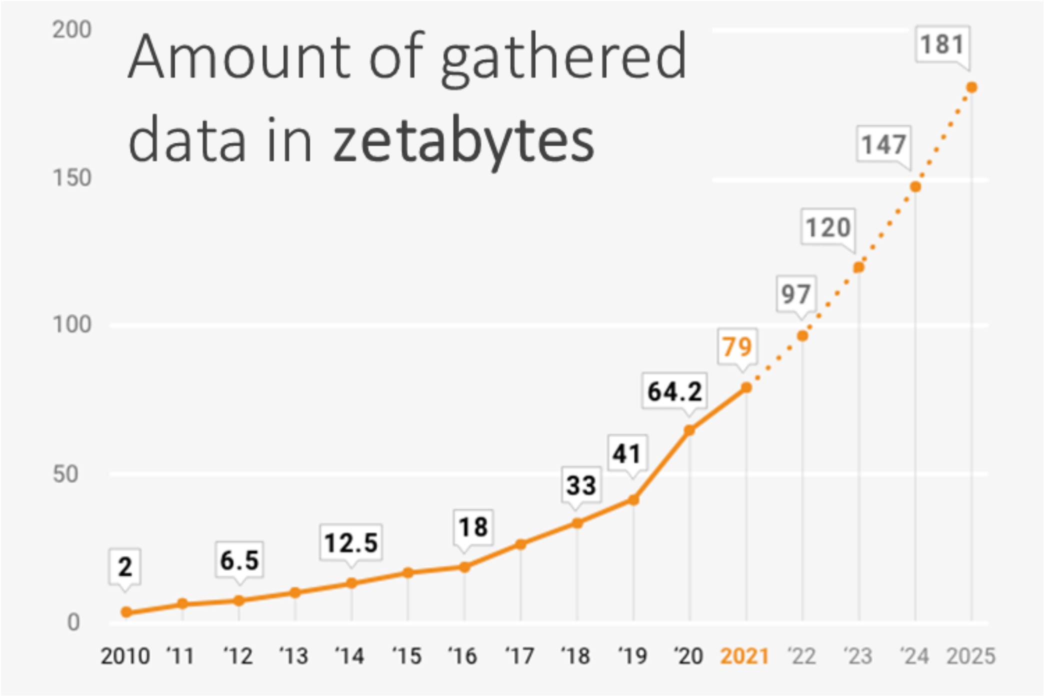


Figure Source: Statista.com

Data is generated from different sources

Efficient pre-processing, cleaning and storing of data became challenging due to generation volume.

How can we manage and store such a huge amount of data?

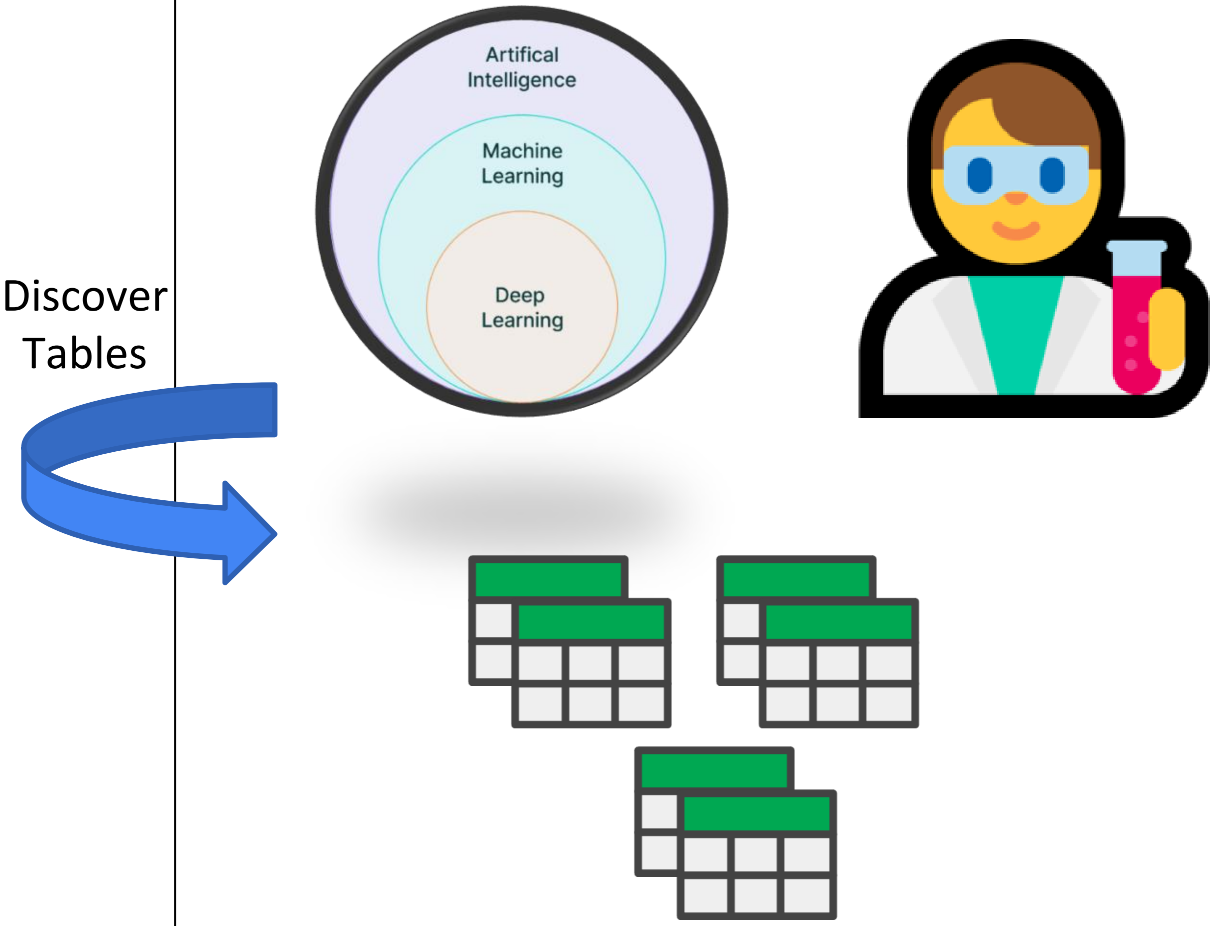
Data Lakes

How can we find the data within data lakes?

- Data lakes: Centralized repositories designed to store large amounts of data.[1]
- Governments release public datasets in open data lakes.
- Enterprises have their own data lakes.

[1] https://cloud.google.com/learn/what-is-a-data-lake

Table Discovery from Data Lakes

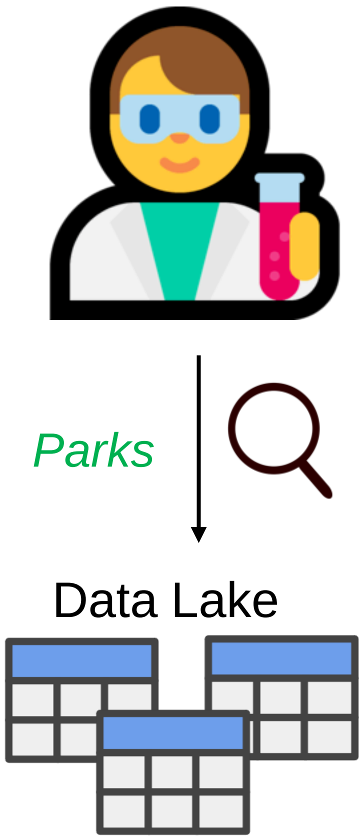


Data lakes contain millions of Tables

Data scientists need a lot of datasets

Data scientists can find the datasets within data lakes and use them for their tasks.

Suppose a data scientist wants to find table about parks



Suppose a data scientist wants to find table about parks

- **Manual Search over whole data lake?**
  - Not possible due to enormous data lake size.

Suppose a data scientist wants to find table about parks

- **Keyword search over metadata:** Discover tables relevant to given keywords [1]
  - Data lakes generally lack proper data semantics (meaning).
  - Misses relevant tables and/or returns irrelevant tables.

Unreliable metadata

Inconsistent values

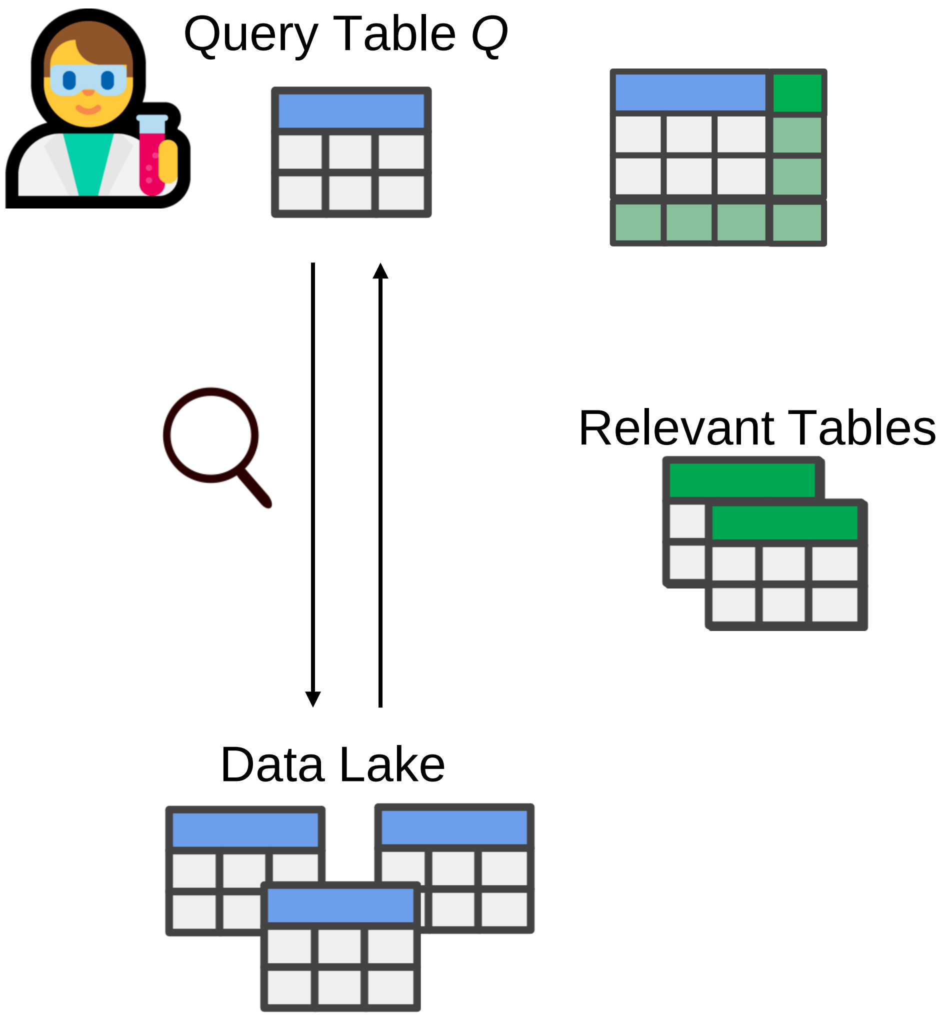
Missing Data

A real data lake table [2]

2017 GradeX Crossings Inventory	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10
1	17568	CN	PAC	BC	Public	F	71.94	Yale		
2	7940	CN	ONT	ON	Public	F	64.74	Kingston - CN		
3	7930	CN	ONT	ON	Public	F	53.82	Kingston - CN		
4	22103	CP	ONT	Ontario	Public	F	17.35	Galt		
5	30450	CP	PNR	SK	Public	F	3.37	Wilkie		
6	7917	CN	ONT	ON	Public	F	34.72	Kingston - CN		
7	7920	CN	ONT	ON	Public	F	37.54	Kingston - CN		
8	5039	CN	ONT	ON	Public	F	77.36	Dundas		
9	5048	CN	ONT	ON	Public	F	77.51	Dundas		
10	7902	CN	ONT	ON	Public	F	17.52	Kingston - CN		
11	35213	CP	PNR	MB	Public	F	4.54	Emerson		
12	4861	CN	QUE	QC	Public	F	117.22	Drummondville		
13	14506	VIA	ONT	ON	Public	F	3.26	Smiths Falls		

[1] Brickley, Burgess and Noy. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. WWW 2019
[2] Nargesian, Zhu, Pu and Miller. Table Union Search on Open Data. PVLDB 2018

Table as a Query



Automate Retrieval of Relevant Tables to a Query Table

Table Discovery[1]

Joinable Table Search[3,4,5]

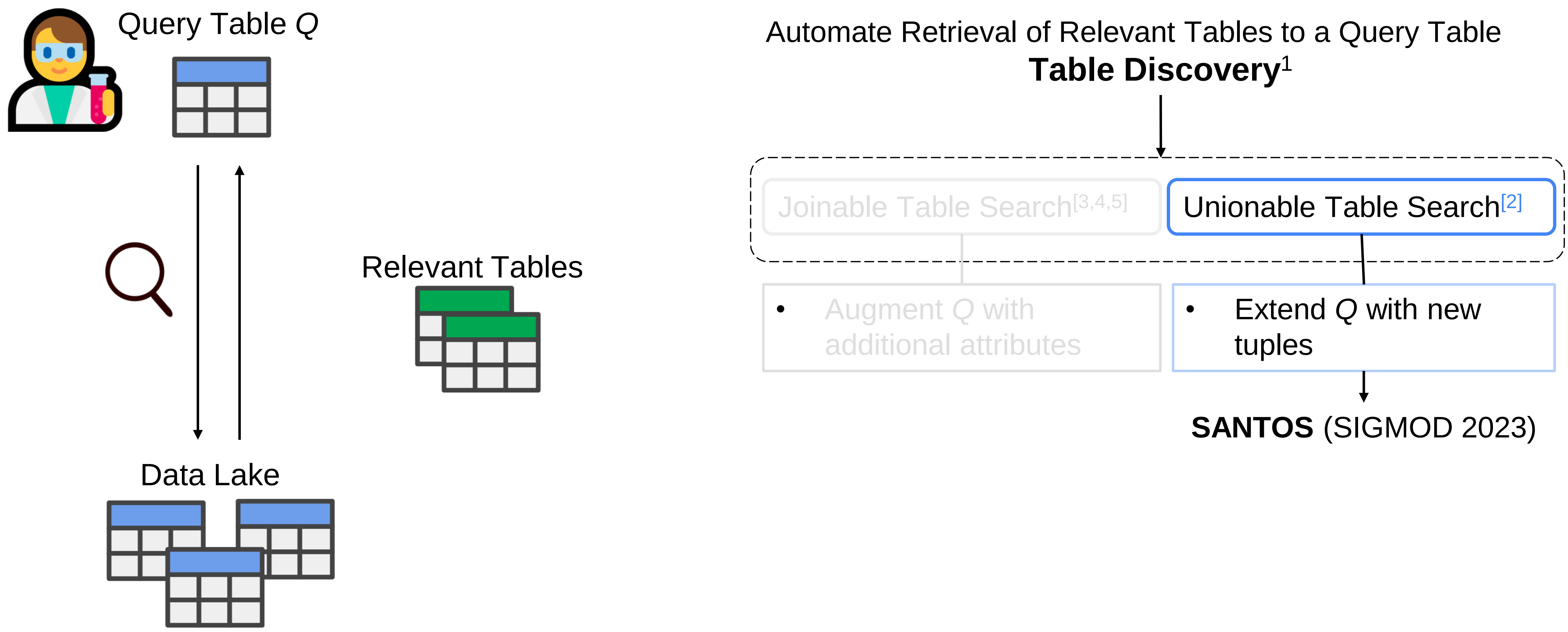
- Augment *Q* with additional attributes

Unionable Table Search[2]

- Extend *Q* with new tuples

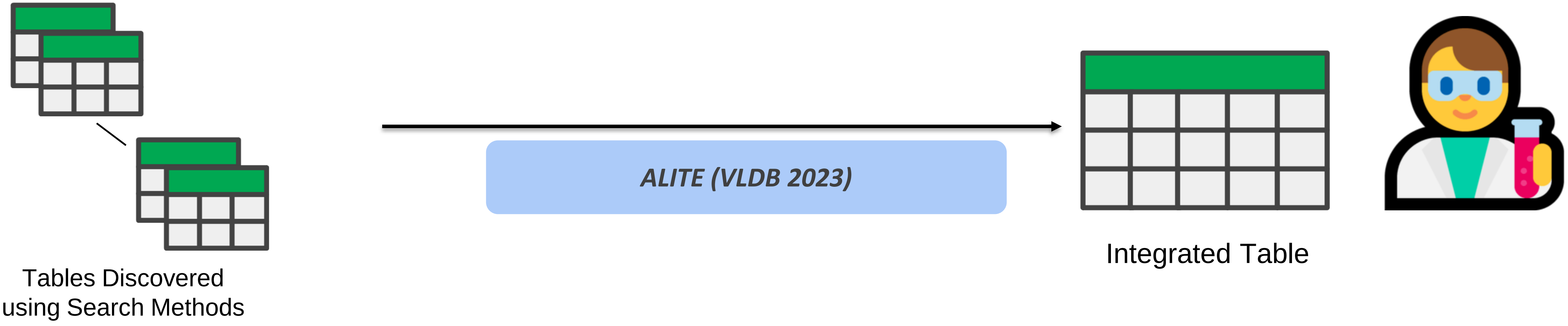
[1] Castelo, Rampin, Santos, Bessa, Chirigati and Freire. Auctus: A dataset search engine for data discovery and augmentation. PVLDB 2021
[2] Nargesian, Zhu, Pu and Miller. Table Union Search on Open Data. PVLDB 2018
[3] Santos, Bessa, Musco and Freire. A Sketch-based Index for Correlated Dataset Search. ICDE 2022
[4] Zhu, Deng, Nargesian and Miller. JOSIE: Overlap set similarity search for finding joinable tables in data lakes. SIGMOD 2019
[5] Zhu, Nargesian, Pu and Miller. LSH ensemble: Internet-scale domain search. PVLDB 2016

Table as a Query



[1] Castelo, Rampin, Santos, Bessa, Chirigati and Freire. Auctus: A dataset search engine for data discovery and augmentation. PVLDB 2021
[2] Nargesian, Zhu, Pu and Miller. Table Union Search on Open Data. PVLDB 2018
[3] Santos, Bessa, Musco and Freire. A Sketch-based Index for Correlated Dataset Search. ICDE 2022
[4] Zhu, Deng, Nargesian and Miller. JOSIE: Overlap set similarity search for finding joinable tables in data lakes. SIGMOD 2019
[5] Zhu, Nargesian, Pu and Miller. LSH ensemble: Internet-scale domain search. PVLDB 2016

Integrating the Discovered Tables



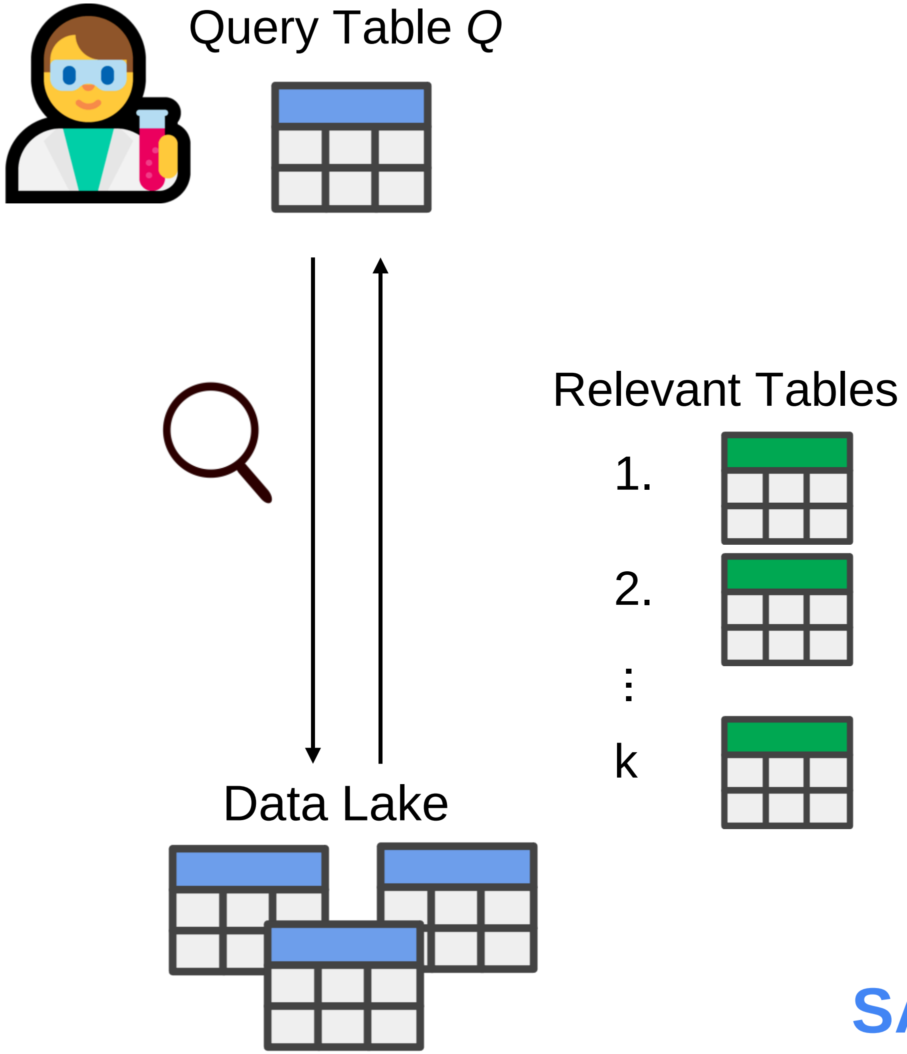
- **How can we integrate the discovered tables into a single table?**
  - Integration provides a unified view of data.
  - Integration allows data scientists to run queries that go beyond a single table.

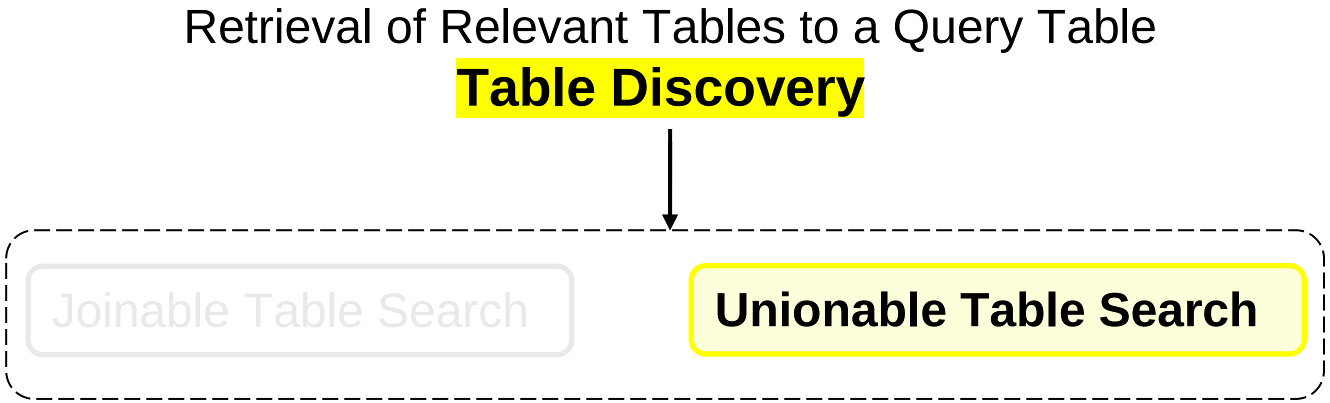
We present ALITE as a solution to integrate the set of discovered tables.

Outline

- Motivation
- **Table discovery using SANTOS**
- Table integration using ALITE
- DIALITE

SANTOS: Relationship-based Semantic Table Union Search





SANTOS finds top-k semantically unionable tables for a given query table.

SANTOS: Relationship-based Semantic Table Union Search

Data Scientist's table (Query Table)

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

(a) A table about parks

Data lake tables (Candidate unionable Tables)

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. 64th St.	773 284-7328	Riverside	Adrian Molina	Pixar
-----	------	-------	------	----	------	------

(b) A table about films shown in different parks

Person	Occupation	Birthplace	Country	Park Name
James Taylor	Singer	Boston	USA	Central Park
Anthony Pelissier	Film Director	Barnet	UK	Cairngorms National Park
Akram Afif	Football Player	Doha	Qatar	Aspire Park
Ivan A. Getting	Physicist	NYC	USA	El Segundo Park
Abby May	Social Worker	Boston	USA	Fenway Park
Stevie Ray Vaughan	Singer	Texas	USA	Chastain Park

(c) A table about people

- Assume, the data scientist is collecting Park information for certain analysis.
- The data scientist wants to add rows (i.e., unionable tables).

Metadata may be inconsistent or imprecise!

Existing Methods

Data Scientist's table (Query Table)

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

(a) A table about parks

Data lake tables (Candidate unionable Tables)

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. 64th St.	773 284-7328	Riverside	Adrian Molina	Pixar
-----	------	-------	------	----	------	------

(b) A table about films shown in different parks

Person	Occupation	Birthplace	Country	Park Name
James Taylor	Singer	Boston	USA	Central Park
Anthony Pelissier	Film Director	Barnet	UK	Cairngorms National Park
Akram Afif	Football Player	Doha	Qatar	Aspire Park
Ivan A. Getting	Physicist	NYC	USA	El Segundo Park
Abby May	Social Worker	Boston	USA	Fenway Park
Stevie Ray Vaughan	Singer	Texas	USA	Chastain Park

(c) A table about people

Existing Methods [1,2]:

- Look for unionable columns.
- Higher the number of unionable columns with better match, better is the table unionability.

Table b: two unionable columns (Park Name ∪ Park Name, City ∪ Park City)

Table c: four unionable columns (Supervisor ∪ Person, City ∪ Birthplace, Country ∪ Country, Park Name ∪ Park Name)

Hence, Table (c) is considered as the better match!!!

[1] Nargesian, Zhu, Pu and Miller. Table Union Search on Open Data. PVLDB 2018
[2] Bogatu, Fernandes, Paton and Konstantinou. Dataset Discovery in Data Lakes. ICDE 2020

Existing Methods

Data Scientist's table (Query Table)

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

(a) A table about parks

Data lake tables (Candidate unionable Tables)

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) A table about films shown in different parks

Person	Occupation	Birthplace	Country	Park Name
James Taylor	Singer	Boston	USA	Central Park
Anthony Pelissier	Film Director	Barnet	UK	Cairngorms National Park
Akram Afif	Football Player	Doha	Qatar	Aspire Park
Ivan A. Getting	Physicist	NYC	USA	El Segundo Park
Abby May	Social Worker	Boston	USA	Fenway Park
Stevie Ray Vaughan	Singer	Texas	USA	Chastain Park

(c) A table about people

Table (c) is not primarily about parks.

Existing Methods

Data Scientist's table (Query Table)

Park Name	Supervisor	City	Country
River Park	Vera Onate	Fresno	USA
West Lawn Park	Paul Veliotis	Chicago	USA
Central Park	James Taylor	Boston	USA
Cairngorms National Park	Anthony Pelissier	Barnet	UK
-------	-------	------	----------

(a) A table about parks

Data lake tables (Candidate unionable Tables)

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. 64th St.	773 284-7328	Riverside	Adrian Molina	Pixar
-----	------	-------	------	----	------	------

(b) A table about films shown in different parks

False unioning
adds erroneous tuples!!

Person	Occupation	Birthplace	Country	Park Name
James Taylor	Singer	Boston	USA	Central Park
Anthony Pelissier	Film Director	Barnet	UK	Cairngorms National Park
Akram Afif	Football Player	Doha	Qatar	Aspire Park
Ivan A. Getting	Physicist	NYC	USA	El Segundo Park
Abby May	Social Worker	Boston	USA	Fenway Park
Stevie Ray Vaughan	Singer	Texas	USA	Chastain Park

(c) A table about people

Table (c) is not primarily about parks.

Hence, Column semantics is not enough to infer table unionability.

SANTOS Model

Data Scientist's table (Query Table)

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

(a) A table about parks

Data lake tables (Candidate unionable Tables)

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. 64th St.	773 284-7328	Riverside	Adrian Molina	Pixar
-----	-----	-------	------	----	------	------

(b) A table about films shown in different parks

Person	Occupation	Birthplace	Country	Park Name
James Taylor	Singer	Boston	USA	Central Park
Anthony Pelissier	Film Director	Barnet	UK	Cairngorms National Park
Akram Afif	Football Player	Doha	Qatar	Aspire Park
Ivan A. Getting	Physicist	NYC	USA	El Segundo Park
Abby May	Social Worker	Boston	USA	Fenway Park
Stevie Ray Vaughan	Singer	Texas	USA	Chastain Park

(c) A table about people

"Along with column semantics,

we consider the **binary relationships** between the column pairs."

* For conciseness, the lines showing the binary relationships are superimposed in Table (b) and Table (c).

Contributions

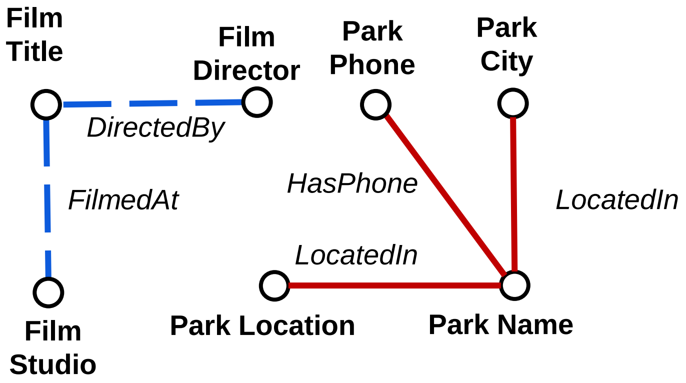
- SANTOS: a new technique for table union search that leverages semantics of columns **and binary relationships** between columns.
- SANTOS uses both an external KB and a **novel data-driven synthesized KB** to find column and relationship semantics.

SANTOS Model

- **We represent all tables in the form of semantic graphs.**
  - Nodes: Columns
  - Edges: Relationships between the columns

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. 64th St.	773 284-7328	Riverside	Adrian Molina	Pixar
-----	------	-------	------	----	------	------

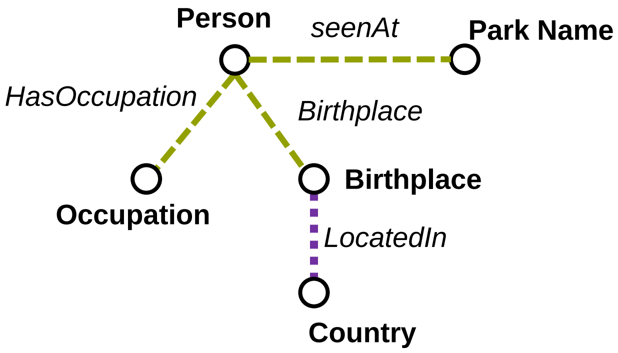
(a) A table about films shown in different parks



(b) Semantic graph of Table (a)

Person	Occupation	Birthplace	Country	Park Name
James Taylor	Singer	Boston	USA	Central Park
Anthony Pelissier	Film Director	Barnet	UK	Cairngorms National Park
Akram Afif	Football Player	Doha	Qatar	Aspire Park
Ivan A. Getting	Physicist	NYC	USA	El Segundo Park
Abby May	Social Worker	Boston	USA	Fenway Park
Stevie Ray Vaughan	Singer	Texas	USA	Chastain Park

(c) A table about People



(d) Semantic graph of Table (c)

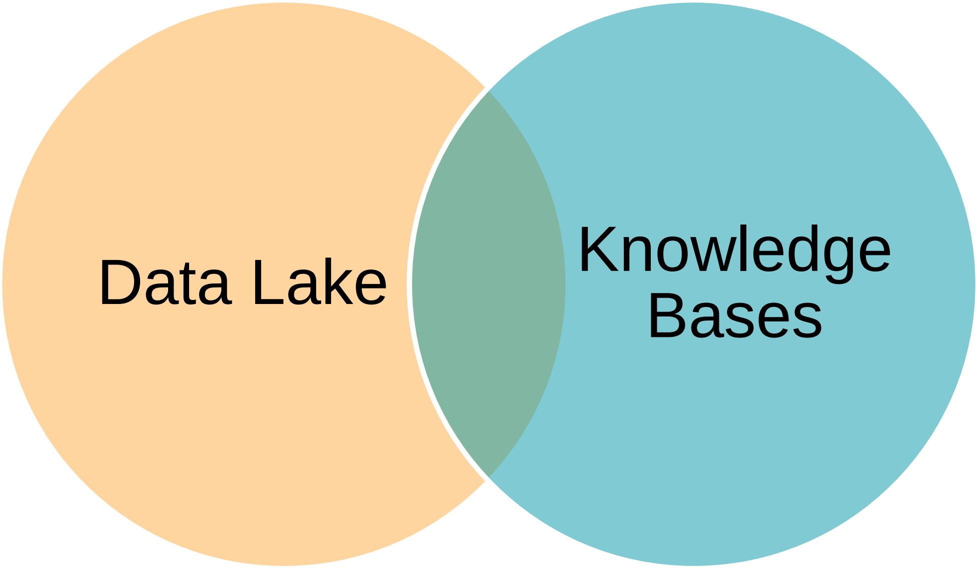
Unionability considers both column semantics and the semantics of binary relationship between the columns.

Creation of Semantic Graph

- **Using External Knowledge Base (YAGO):**
  - Nodes: Types in the Knowledge Bases (KBs)
  - Edges: Relationships in the KBs
  - Each node and edge are assigned with confidence scores.

Knowledge Base Coverage Problem

- Knowledge Bases cover limited entities in the Data lake.
- Hence, relying solely on them is not effective.
- **Solution:** Create a data driven synthesized Knowledge base using the data lake itself.



Creating Synthesized Semantic Graph

- **Synthesized Column Semantics**
  - **Assumption 1:**
    - All values in a column are of same types
  - **Assumption 2:**
    - Two columns are possibly of the same type if they have overlapping values
  - **Concept:**
    - Assign a synthesized column semantics to each column even if we do not know their names
  - Assign a confidence score to each type.

Table 1

A
Brands Park
Kells Park
Eckhart Park

Table 2

D
Kells Park
Eckhart Park
Union Park
Chopin Park
Wicker Park

Table 3

F
Union Park
Gill Park

Creating Synthesized Semantic Graph

- **Synthesized Relationship Semantics**
  - **Assumption 1:**
    - The column pairs in a functional relationship have a possible relationship
  - **Assumption 2:**
    - Two column pairs having overlapping value pairs possibly have same relationship
  - **Concept:**
    - Assign a synthesized relationship semantics to each **column pair in a functional relationship** even if we do not know their names
  - Assign a confidence score to each relationship.

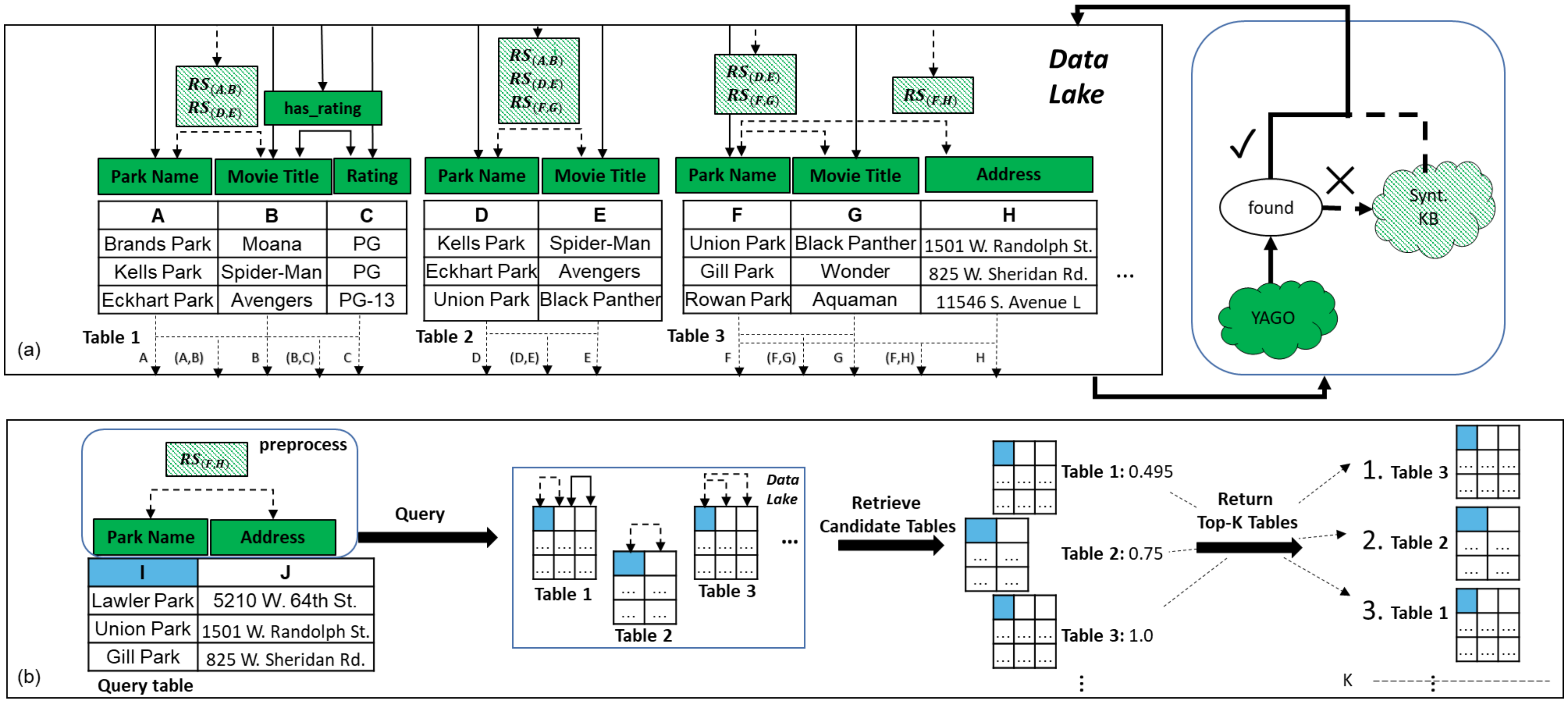
Table 1

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

Table 2

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

SANTOS overall Pipeline



Offline Phase

Online Phase

- **Offline phase:** Create semantic graphs for data lake tables and index them.
- **Online phase:** Find column semantics and relationship semantics for query table and then query the index to find top-k unionable data lake tables.

Experimental Setup: Major Questions

1. How important is the relationship semantics in searching for the top-k unionable tables?
2. How important are SANTOS components in searching for the top-k unionable tables?
3. How scalable is SANTOS in searching for the top-k unionable tables?

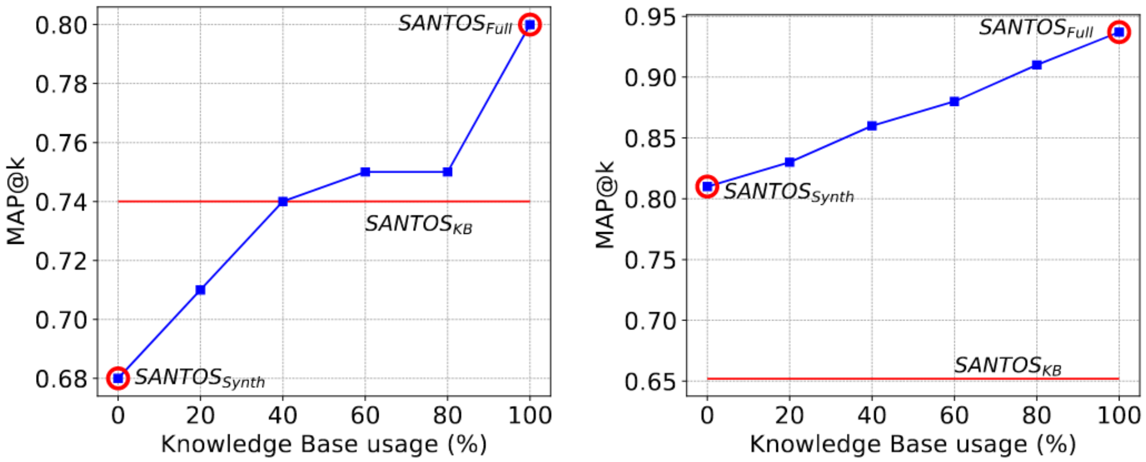
Experimental Setup: Baselines

- **D^{3}L [2]**
  - Column-based Baseline: find related tables based on 5 metrics
  - Column names, value overlap, formatting, word embeddings, domain distributions
- **TURL [3]**
  - Baseline: representation learning over web tables for column type annotation and relation extraction tasks

[1] Nargesian, Zhu, Pu and Miller. Table Union Search on Open Data. PVLDB 2018
[2] Bogatu, Fernandes, Paton and Konstantinou. Dataset Discovery in Data Lakes. ICDE 2020
[3] Deng, Sun, Lees, Wu and Yu. TURL: Table Understanding Through Representation Learning. VLDB 2021

Experimental Results

1. SANTOS outperforms the state-of-the-art method [2] by 25-165% in Mean Average Precision (MAP) across all benchmarks.
2. Synthesized KB improves MAP by 8% on TUS benchmark [1] and 43% on **(New)** SMALL benchmark .



(a) MAP@60 on TUS (b) MAP@10 on SMALL

Efficiency

3. SANTOS's query time on **(New)** LARGE benchmark is ~5X faster than state-of-the-art method.

[1] Nargesian, Zhu, Pu and Miller. Table Union Search on Open Data. PVLDB 2018
[2] Bogatu, Fernandes, Paton and Konstantinou. Dataset Discovery in Data Lakes. ICDE 2020
[3] Deng, Sun, Lees, Wu and Yu. TURL: Table Understanding Through Representation Learning. VLDB 2021

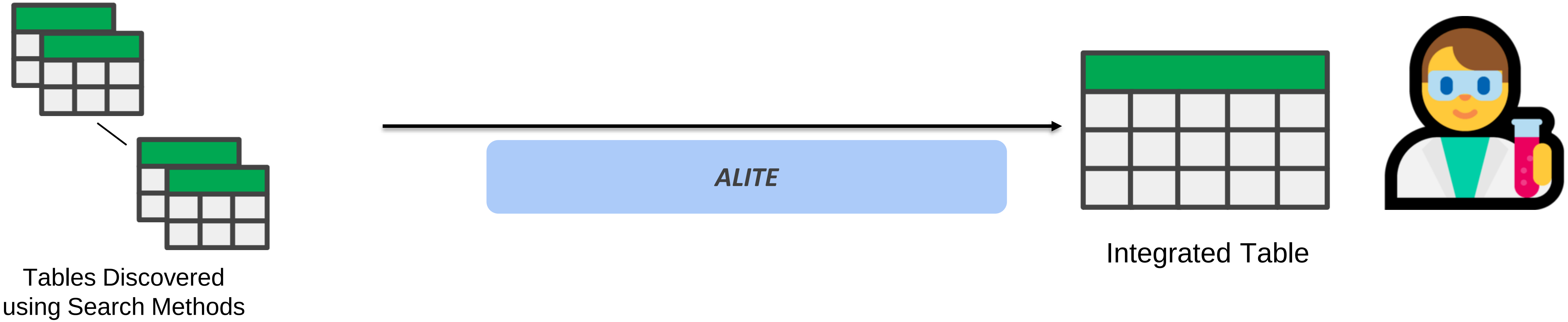
SANTOS Summary

- SANTOS leverages semantics of both columns and relationships between columns for table union search
- SANTOS uses both an external KB and a novel data-driven synthesized KB to find column and relationship semantics
- SANTOS outperforms state-of-the-art table union search method on all benchmarks

Outline

- Motivation
- Table discovery using SANTOS
- **Table integration using ALITE**
- DIALITE

Integrating the Discovered Tables



- **How can we integrate the discovered tables into a single table?**
  - Integration provides a unified view of data.
  - Integration allows data scientists to run queries that go beyond a single table.

We present ALITE as a solution to integrate the set of discovered tables.

Why Table Integration?

T_1

TID	Stadium	Location	Team
t_1	NRG Stadium	Texas	Houston Texans
t_2	AT&T Stadium	Texas	Dallas Cowboys
t_3	Paul Brown	Ohio	±
t_4	Sofi Stadium	California	Angeles Chargers

T_2

TID	Stadium	Location	Opened
t_5	Soldier Field	Chicago	1924
t_6	Ford Field	Michigan	2002

T_3

TID	Team	Location	Coach
t_7	Houston Texans	Texas	Lovie Smith
t_8	Green Bay Packers	Wisconsin	Matt LaFleur
t_9	Detroit Lions	Michigan	Dan Campbell

T_4

TID	Stadium	Location	Capacity
t_{10}	NRG Stadium	Texas	±
t_{11}	Ford Field	Michigan	65k

T_5

TID	Stadium	Location	Team
t_{12}	Lambeau Field	Wisconsin	Green Bay Packers
t_{13}	±	Ohio	Cleveland
t_{14}	Sofi Stadium	California	±

Null value

Figure. Collected Tables about football stadiums to be integrated*

- **Example query:**
  - *"Find the coaches who coach teams having stadiums established after 2000, that accommodate at least 50 thousand spectators."*

* The collected tables can have more columns; for conciseness, we only show the columns that we use for the discussion.
* TID stands for Tuple IDs. They are not real columns, and we use them for representation only.

Why Table Integration?

T_1

TID	Stadium	Location	Team
t_1	NRG Stadium	Texas	Houston Texans
t_2	AT&T Stadium	Texas	Dallas Cowboys
t_3	Paul Brown	Ohio	±
t_4	Sofi Stadium	California	Angeles Chargers

T_2

TID	Stadium	Location	Opened
t_5	Soldier Field	Chicago	1924
t_6	Ford Field	Michigan	2002

T_3

TID	Team	Location	Coach
t_7	Houston Texans	Texas	Lovie Smith
t_8	Green Bay Packers	Wisconsin	Matt LaFleur
t_9	Detroit Lions	Michigan	Dan Campbell

T_4

TID	Stadium	Location	Capacity
t_{10}	NRG Stadium	Texas	±
t_{11}	Ford Field	Michigan	65k

T_5

TID	Stadium	Location	Team
t_{12}	Lambeau Field	Wisconsin	Green Bay Packers
t_{13}	±	Ohio	Cleveland
t_{14}	Sofi Stadium	California	±

Null value

Figure. Collected Tables about football stadiums to be integrated*

- **Example query:**
  - *"Find the coaches who coach teams having stadiums established after 2000, that accommodate at least 50 thousand spectators."*
  - *Dan Campbell is a coach who coaches the Detroit Lions that uses Ford Field Stadium established in 2002 and having a capacity of hosting 65k spectators.*

We need to (at least) integrate Tables T_2, T_3 and T_4 to get this answer.

* The collected tables can have more columns; for conciseness, we only show the columns that we use for the discussion.
* TID stands for Tuple IDs. They are not real columns, and we use them for representation only.

Issues with Table Integration:

- Which columns to align together? Meta data could be imprecise!

Different column name, same semantics!

T_2

TID	Name	Location	Opened
t_5	Soldier Field	Chicago	1924
t_6	Ford Field	Michigan	2002

T_3

TID	Name	State	Coach
t_7	Houston Texans	Texas	Lovie Smith
t_8	Green Bay Packers	Wisconsin	Matt LaFleur
t_9	Detroit Lions	Michigan	Dan Campbell

Same column name, different semantics!

Figure. Collected Tables about football stadiums to be integrated

Issues with Table Integration

- Which columns to align together?
  - Schema matching between a table pair is not enough as we want to integrate a set of tables.

T_1

TID	Stadium	Location	Team
t_1	NRG Stadium	Texas	Houston Texans
t_2	AT&T Stadium	Texas	Dallas Cowboys
t_3	Paul Brown	Ohio	±
t_4	Sofi Stadium	California	Angeles Chargers

T_2

TID	Stadium	Location	Opened
t_5	Soldier Field	Chicago	1924
t_6	Ford Field	Michigan	2002

T_3

TID	Team	Location	Coach
t_7	Houston Texans	Texas	Lovie Smith
t_8	Green Bay Packers	Wisconsin	Matt LaFleur
t_9	Detroit Lions	Michigan	Dan Campbell

T_4

TID	Stadium	Location	Capacity
t_{10}	NRG Stadium	Texas	±
t_{11}	Ford Field	Michigan	65k

T_5

TID	Stadium	Location	Team
t_{12}	Lambeau Field	Wisconsin	Green Bay Packers
t_{13}	±	Ohio	Cleveland
t_{14}	Sofi Stadium	California	±

Figure. Collected Tables about football stadiums to be integrated

We develop a hierarchical clustering algorithm that determines the aligning columns using holistic schema matching [1].

[1] He and Chang. Making holistic schema matching robust: an ensemble approach. KDD 2005

Issues with Table Integration

- Basic integration operators may not be effective.

T_1

TID	Stadium	Location	Team
t_1	NRG Stadium	Texas	Houston Texans
t_2	AT&T Stadium	Texas	Dallas Cowboys
t_3	Paul Brown	Ohio	±
t_4	Sofi Stadium	California	Angeles Chargers

T_2

TID	Stadium	Location	Opened
t_5	Soldier Field	Chicago	1924
t_6	Ford Field	Michigan	2002

T_3

TID	Team	Location	Coach
t_7	Houston Texans	Texas	Lovie Smith
t_8	Green Bay Packers	Wisconsin	Matt LaFleur
t_9	Detroit Lions	Michigan	Dan Campbell

T_4

TID	Stadium	Location	Capacity
t_{10}	NRG Stadium	Texas	±
t_{11}	Ford Field	Michigan	65k

T_5

TID	Stadium	Location	Team
t_{12}	Lambeau Field	Wisconsin	Green Bay Packers
t_{13}	±	Ohio	Cleveland
t_{14}	Sofi Stadium	California	±

Figure. Collected Tables about football stadiums to be integrated

- **Union operator (∪):**
  - Needs all tables to have the exact same columns.
  - We can project out the aligning columns, but we miss other columns.

$T_1 \cup T_2 \cup T_3 \cup T_4 \cup T_5$

Location
Texas
Ohio
California
Chicago
Michigan
Wisconsin

Issues with Table Integration

- Basic integration operators may not be effective.

T_1

TID	Stadium	Location	Team
t_1	NRG Stadium	Texas	Houston Texans
t_2	AT&T Stadium	Texas	Dallas Cowboys
t_3	Paul Brown	Ohio	±
t_4	Sofi Stadium	California	Angeles Chargers

T_2

TID	Stadium	Location	Opened
t_5	Soldier Field	Chicago	1924
t_6	Ford Field	Michigan	2002

T_3

TID	Team	Location	Coach
t_7	Houston Texans	Texas	Lovie Smith
t_8	Green Bay Packers	Wisconsin	Matt LaFleur
t_9	Detroit Lions	Michigan	Dan Campbell

T_4

TID	Stadium	Location	Capacity
t_{10}	NRG Stadium	Texas	±
t_{11}	Ford Field	Michigan	65k

T_5

TID	Stadium	Location	Team
t_{12}	Lambeau Field	Wisconsin	Green Bay Packers
t_{13}	±	Ohio	Cleveland
t_{14}	Sofi Stadium	California	±

Figure. Collected Tables about football stadiums to be integrated

- **Inner join (⋈):**
  - Missed tuples having no join partner even in one table.

$$T_1 \bowtie T_2 \bowtie T_3 \bowtie T_4 \bowtie T_5$$

TID	Stadium	Location	Team	Opened	Coach	Capacity

Issues with Table Integration

- Basic integration operators may not be effective.

T_1

TID	Stadium	Location	Team
t_1	NRG Stadium	Texas	Houston Texans
t_2	AT&T Stadium	Texas	Dallas Cowboys
t_3	Paul Brown	Ohio	±
t_4	Sofi Stadium	California	Angeles Chargers

T_2

TID	Stadium	Location	Opened
t_5	Soldier Field	Chicago	1924
t_6	Ford Field	Michigan	2002

T_3

TID	Team	Location	Coach
t_7	Houston Texans	Texas	Lovie Smith
t_8	Green Bay Packers	Wisconsin	Matt LaFleur
t_9	Detroit Lions	Michigan	Dan Campbell

T_4

TID	Stadium	Location	Capacity
t_{10}	NRG Stadium	Texas	±
t_{11}	Ford Field	Michigan	65k

T_5

TID	Stadium	Location	Team
t_{12}	Lambeau Field	Wisconsin	Green Bay Packers
t_{13}	±	Ohio	Cleveland
t_{14}	Sofi Stadium	California	±

Figure. Collected Tables about football stadiums to be integrated

- **Outer join ($⟗$):**
  - Not associative.
  - Different order of integration using outer join could yield different outputs [1].
  - Example: $(T_1 ⟗ T_2 ⟗ T_3 ⟗ T_4 ⟗ T_5) \neq (T_5 ⟗ T_4 ⟗ T_3 ⟗ T_2 ⟗ T_1)$

We propose to use a scalable implementation of (Natural) Full Disjunction operator [1].

[1] Galindo-Legaria. Outerjoins as Disjunctions. SIGMOD 1994

Full Disjunction (FD)

T_1

TID	Stadium	Location	Team
t_1	NRG Stadium	Texas	Houston Texans
t_2	AT&T Stadium	Texas	Dallas Cowboys
t_3	Paul Brown	Ohio	±
t_4	Sofi Stadium	California	Angeles Chargers

T_2

TID	Stadium	Location	Opened
t_5	Soldier Field	Chicago	1924
t_6	Ford Field	Michigan	2002

T_3

TID	Team	Location	Coach
t_7	Houston Texans	Texas	Lovie Smith
t_8	Green Bay Packers	Wisconsin	Matt LaFleur
t_9	Detroit Lions	Michigan	Dan Campbell

T_4

TID	Stadium	Location	Capacity
t_{10}	NRG Stadium	Texas	±
t_{11}	Ford Field	Michigan	65k

T_5

TID	Stadium	Location	Team
t_{12}	Lambeau Field	Wisconsin	Green Bay Packers
t_{13}	±	Ohio	Cleveland
t_{14}	Sofi Stadium	California	±

Figure. Collected Tables about football stadiums to be integrated

$$FD\ (T_1, T_2, T_3, T_4, T_5) = FD\ (T_5, T_4, T_3, T_2, T_1)$$

TIDs	Stadium	Location	Team	Opened	Coach	Capacity
$\{t_1, t_7, t_{10}\}$	NRG Stadium	Texas	Houston Texans	⊥	Lovie Smith	±
$\{t_2\}$	AT&T Stadium	Texas	Dallas Cowboys	⊥	⊥	⊥
$\{t_3\}$	Paul Brown	Ohio	±	⊥	⊥	⊥
$\{t_{13}\}$	±	Ohio	Cleveland	⊥	⊥	⊥
$\{t_4\}$	Sofi Stadium	California	Angeles Chargers	⊥	⊥	⊥
$\{t_5\}$	Soldier Field	Chicago	⊥	1924	⊥	⊥
$\{t_6, t_9, t_{11}\}$	Ford Field	Michigan	Detroit Lions	2002	Dan Campbell	65k
$\{t_8, t_{12}\}$	Lambeau Field	Wisconsin	Green Bay Packers	⊥	Matt LaFleur	⊥

Figure. Output tuples generated using Full Disjunction operator

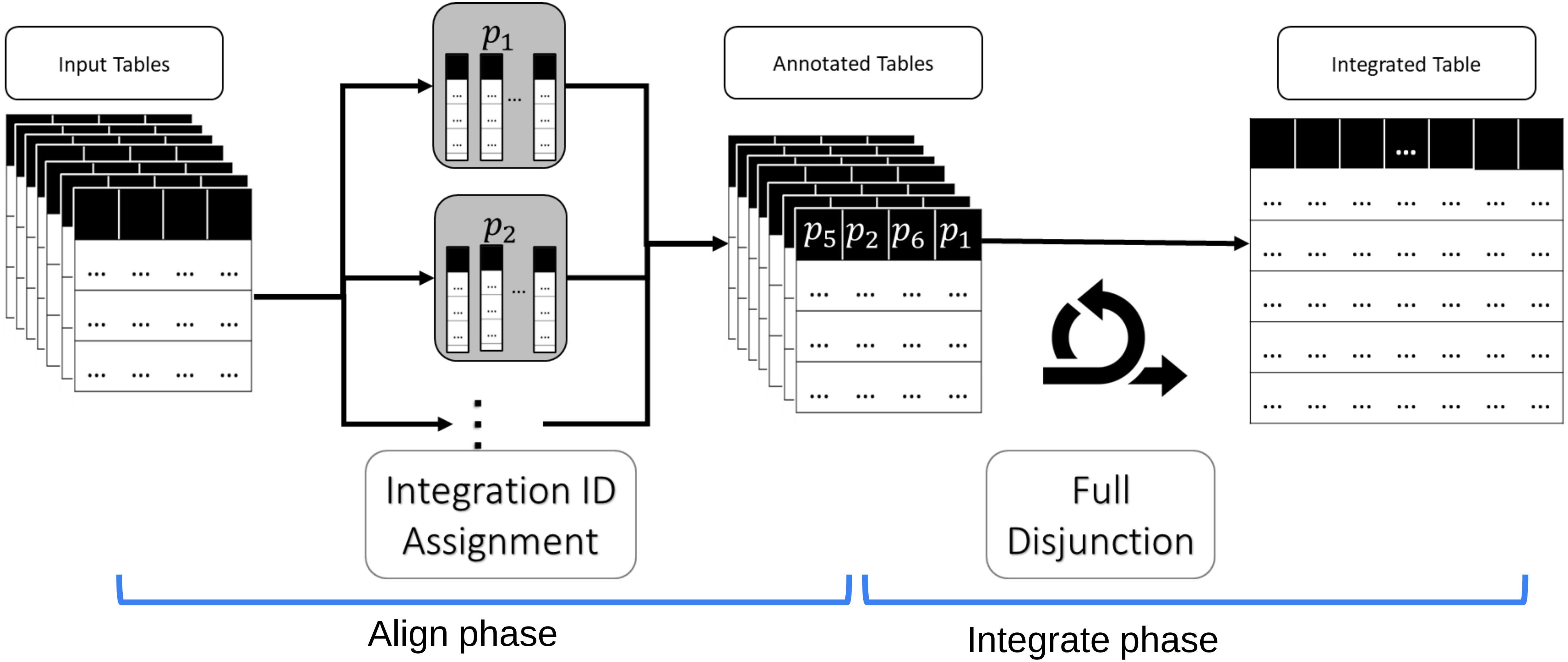
Produced null due to incomplete information

- An associative version of outer join operator [1].
- Integrates each input tuple maximally and produces a set of maximally integrated tuples [2].
- The maximally integrated tuples do not subsume each other.

[1] Galindo-Legaria. Outerjoins as Disjunctions. SIGMOD 1994
[2] Kanza and Sagiv. Computing Full Disjunctions. PODS 2003

Proposed Solution

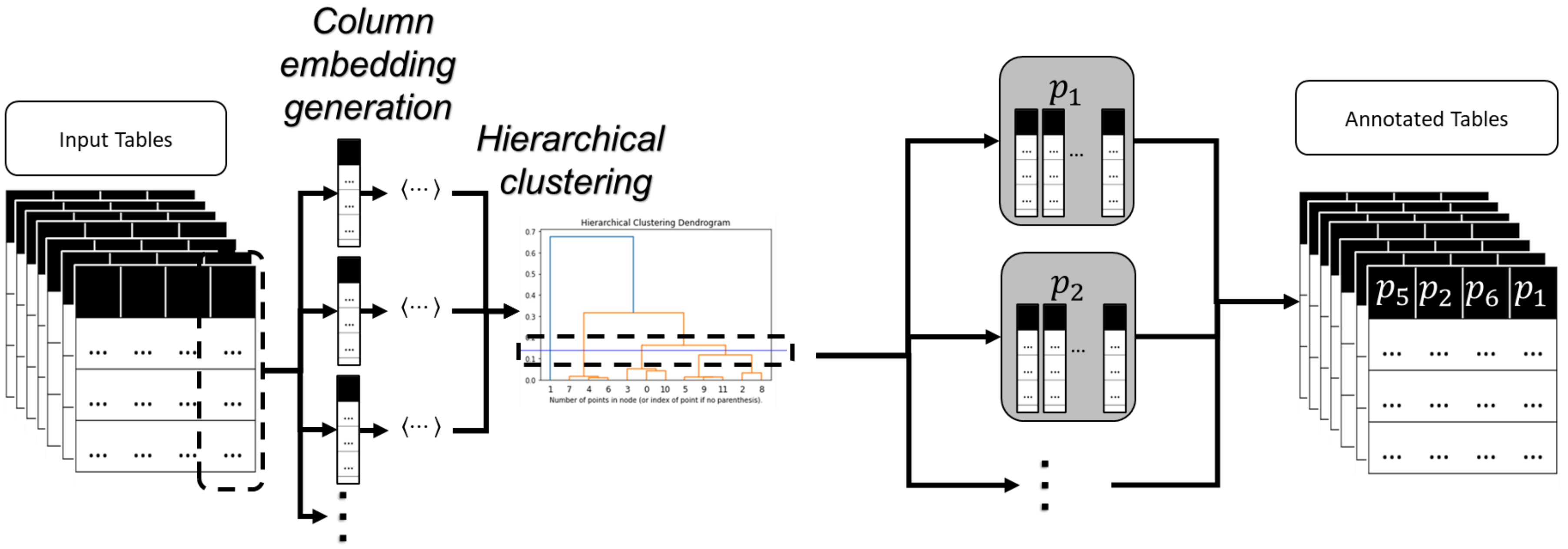
- **ALITE** (**Ali**gn and Integra**te**):



- **Align:** Identify the matching columns across the set of tables and annotate them with a dummy column header.
- **Integrate:** Apply a novel algorithm for Full Disjunction (FD) [1] that scales better than prior work.

[1] Galindo-Legaria. Outerjoins as Disjunctions. SIGMOD 1994

Align Phase (Phase 1)



Selected number of cluster: 6

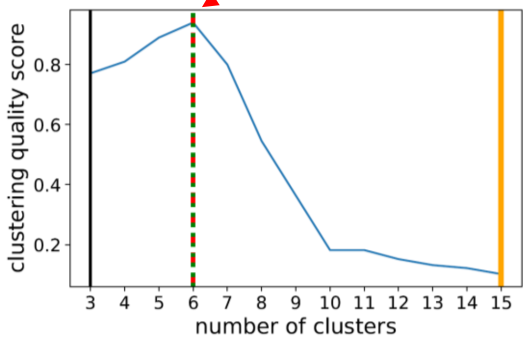
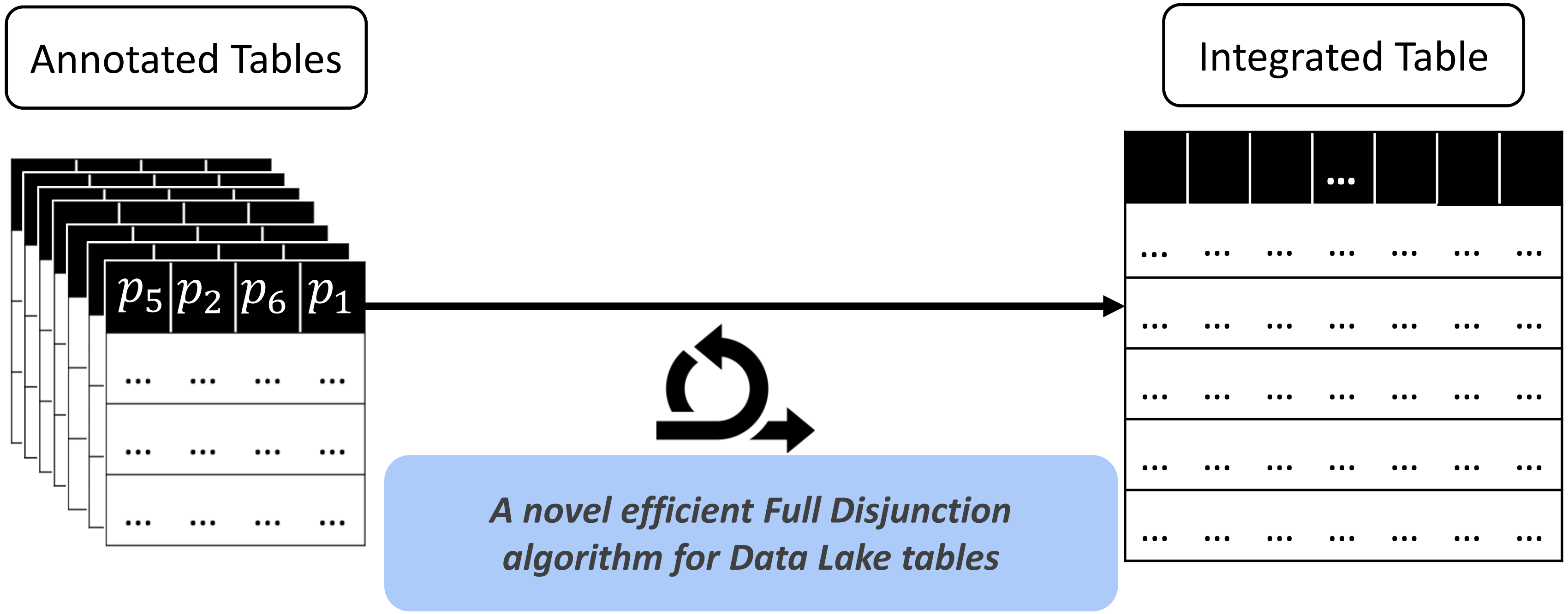


Fig. Number of clusters selection

- **Input:** A set of tables to be integrated.
- **Output:** Different set of columns annotated with their integration IDs.
- **Steps:**
  - Embed each column by using pre-trained embeddings over their values.
  - Apply hierarchical clustering over the columns.
  - Select the number of cluster that maximizes the clustering quality (e.g., Silhouette's coefficient[1]).
  - Annotate each cluster with a dummy column header (column integration ID).

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

Integrate Phase (Phase 2)



- **Input:** A set of tables with their columns annotated with their integration IDs.
- **Output:** An integrated table.

We develop FD algorithm by adopting complementation semantics [5] that practically scales FD computation for table schemas forming complex cycles and having no PK-FK relations.

[1] Galindo-Legaria. Outerjoins as Disjunctions. SIGMOD 1994
[2] Kanza and Sagiv. Computing Full Disjunctions. PODS 2003
[3] Cohen, Fadida, Kanza, Kimelfeld and Sagiv. Full Disjunctions: Poly-Delay Iterator in Action. VLDB 2006
[4] Paganelli, Beneventano, Guerra and Sottovia. Parallelizing Computations of Full Disjunctions. Big Data Research 2019
[5] Bleiholder and Naumann. Data Fusion. CSUR 2009

ALITE Full Disjunction Building Blocks

- **Types of Nulls:**
  - **Missing Nulls**
    - Null values (±) in the input tables.
  - **Produced Nulls**
    - Null values (⊥) produced during the integration process due to incomplete information.

Stadium	Location	Team	Opened	Coach	Capacity
Paul Brown	Ohio	±	⊥	⊥	⊥
±	Ohio	Cleveland	⊥	⊥	⊥
Sofi Stadium	California	Angeles Chargers	⊥	⊥	⊥

Missing Nulls

Produced Nulls

We handle two nulls differently during the integration.

ALITE Full Disjunction Algorithm

- We use a fixed sequence of outer union, complementation and subsumption.
- Produce maximally integrated tuples using complementation operator.
  - Replace missing nulls with distinct labeled nulls to avoid undesirable complementation.
- Remove subsumable tuples using subsumption operator to get FD.

ALITE Full Disjunction

```
Input: 𝒯 = {T_1, T_2, ... T_n}, a set of tables with integration IDs as
  column names
Output: FD(𝒯), the Natural Full Disjunction of 𝒯
𝒯 ← GenerateLabeledNulls(𝒯)
U_ou ← T_1 ⊎ T_2 ⊎ ··· ⊎ T_n                //Apply outer union ⊎
U_comp ← Complement(U_ou)                   //Apply complementation κ
U_comp ← RemoveLabeledNulls(U_comp)
T′ ← β(U_comp)                              //Apply subsumption β
Output T′
```

Key idea: Generating Labeled Nulls
Avoids undesirable complementation

Experiments

- **Benchmark:**
  - We create three new benchmarks using real data lake tables.[1]
  - We also use IMDB movie benchmark.[2]
  - Each integration set contains a set of tables to be integrated together.

Benchmark	Tables	Columns	Tuples	Integration sets	Experiments
Align	606	4,584	2.2M	65	Align
Real	102	1, 195	219k	11	Align, Integrate
Join	302	2, 309	1.1M	28	Integrate
IMDB	6	33	3k - 30k	1	Integrate

[1] https://github.com/northeastern-datalab/alite
[2] https://datasets.imdbws.com

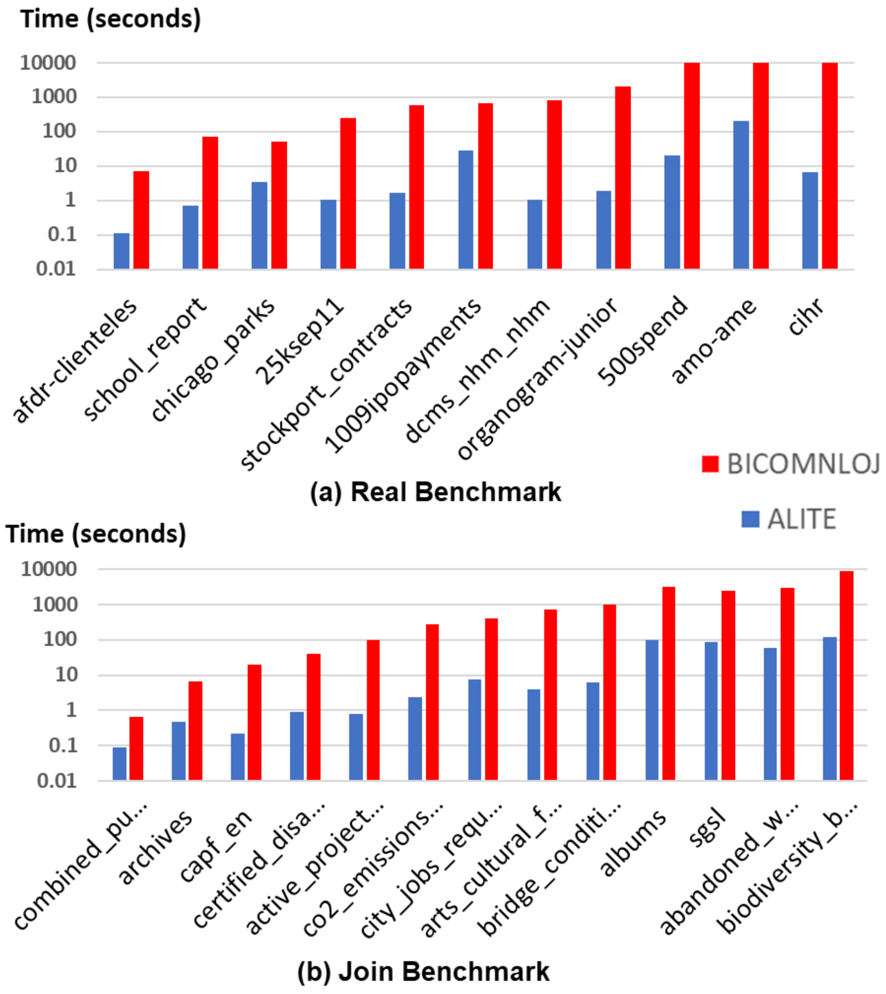
Experiments

- **Align Phase:**
  - **Baselines:**
    - **Schema-matching techniques** [1]
      - We adopt binary matchers such as CUPID, COMA, SF, JLM and DB.
  - **ALITE variations:**
      - We implement ALITE using BERT, fastText and TURL embeddings.
  - **Metrics:**
    - Precision, Recall and F_1-Score
  - **Results:**
    - ALITE based on TURL embeddings, pretrained on tables, outperforms other methods in terms of F_1-Score by over 4 % in Real Benchmark.

[1] https://github.com/delftdata/valentine

Experiments

- **Integrate Phase Efficiency**
  - **Baselines:** BICOMNLOJ [1], ParaFD [2]
  - **Metrics:** Runtime
  - **Results:**
    - ALITE is faster than the best baseline (BICOMNLOJ) by around 10 times in average in Real and join benchmarks.



[1] Cohen, Fadida, Kanza, Kimelfeld and Sagiv. Full Disjunctions: Poly-Delay Iterator in Action. VLDB 2006
[2] Paganelli, Beneventano, Guerra and Sottovia. Parallelizing Computations of Full Disjunctions. Big Data Research 2019

Experiments

- **Integrate Phase Effectiveness**
  - **Baseline:** Outer join
  - **Tuple Difference Ratio:**
    - **Tuple Difference Ratio,** $\mathbf{TDR} = \frac{|\boldsymbol{FD\ Tuples} \cap \boldsymbol{Outer\ join\ Tuples}|}{|\boldsymbol{FD\ Tuples}|}$
    - In Real Benchmark, outer join **correctly** produces all maximally integrated tuples only in one case
    - In other cases, outer join produces more than 50% maximally integrated tuples at least three times.

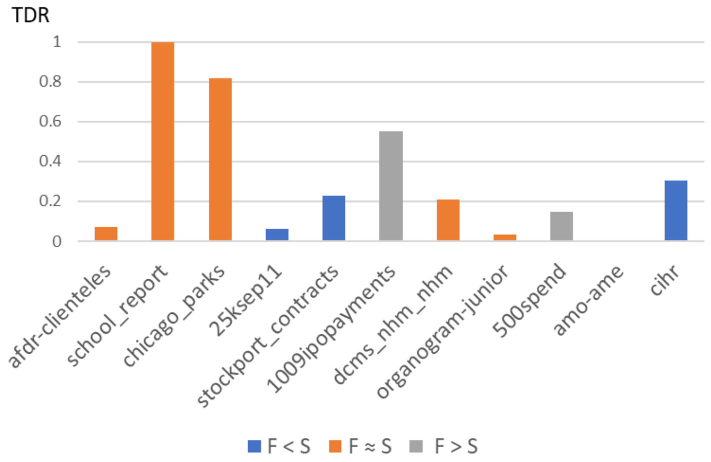


Figure. TDR in Real Benchmark

Experiments

- **Integrate Phase Effectiveness**
  - **Baseline:** Outer join
  - **Entity resolution:**
    - We apply entity resolution to the output of both FD and outer join.
    - Entity resolution over Full disjunction result improves F_1-Score by around 45%.

Integration Method	Precision	Recall	F_1-Score
Full Disjunction	**0.795**	**0.838**	**0.816**
Outer join	0.339	0.397	0.366

Figure. Effectiveness of applying Entity Resolution as a downstreaming task after integration

ALITE Summary

- ALITE outperforms existing schema matchers[1] in Align Phase in terms of F_1-score by over 4 % in Real data lake Benchmark.
- ALITE is faster than the best baseline (BICOMNLOJ [1]) by around 10 times in average in Real benchmarks.
- Entity resolution over Full disjunction result improves F_1-Score by around 45% in comparison to Outer join result.

1 https://github.com/delftdata/valentine

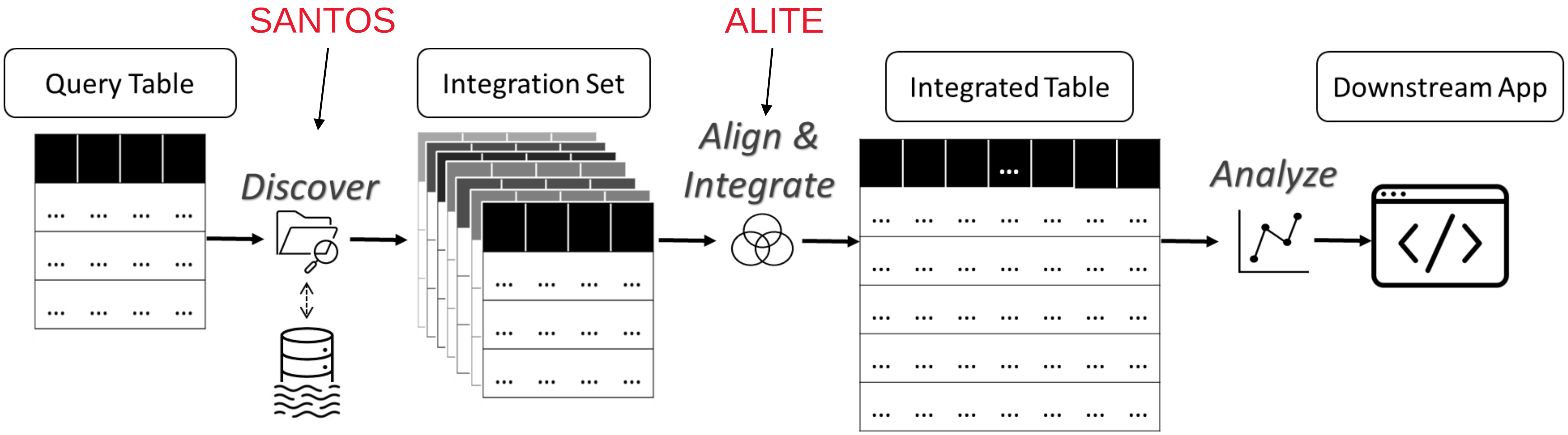
[1] Cohen, Fadida, Kanza, Kimelfeld and Sagiv. Full Disjunctions: Poly-Delay Iterator in Action. VLDB 2006

Outline

- Motivation
- Table discovery using SANTOS
- Table integration using ALITE
- **DIALITE**

End-to-end system

- **DIALITE** (Discover, Align and Integrate)



An overview of DIALITE system

- A system to extend query table by discovering new tables, integrating them.
- DIALITE allows downstreaming task over the integrated table.
- DIALITE is extendible i.e., new discovery and integration algorithms and analyses can be added easily.

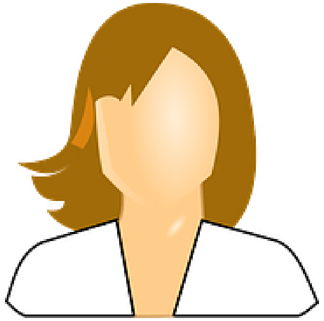
Demonstration

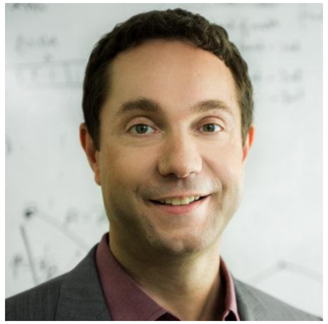
https://tinyurl.com/dialite-sigmod

Acknowledgements (non-exhaustive)

DATALAB

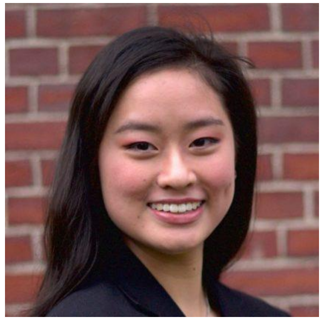

Wolfgang Gatterbauer


Renée J. Miller


Mirek Riedewald


Roee Shraga


Zixuan Chen


Grace Fan

Acknowledgements (non-exhaustive)

Also, thanks to my data scientist ☺

Table Discovery and Integration in Data Lakes

Aamod Khatiwada

Northeastern University

Khoury College of Computer Sciences

Boston

khatiwada.a@northeastern.edu | https://aamodkh.github.io