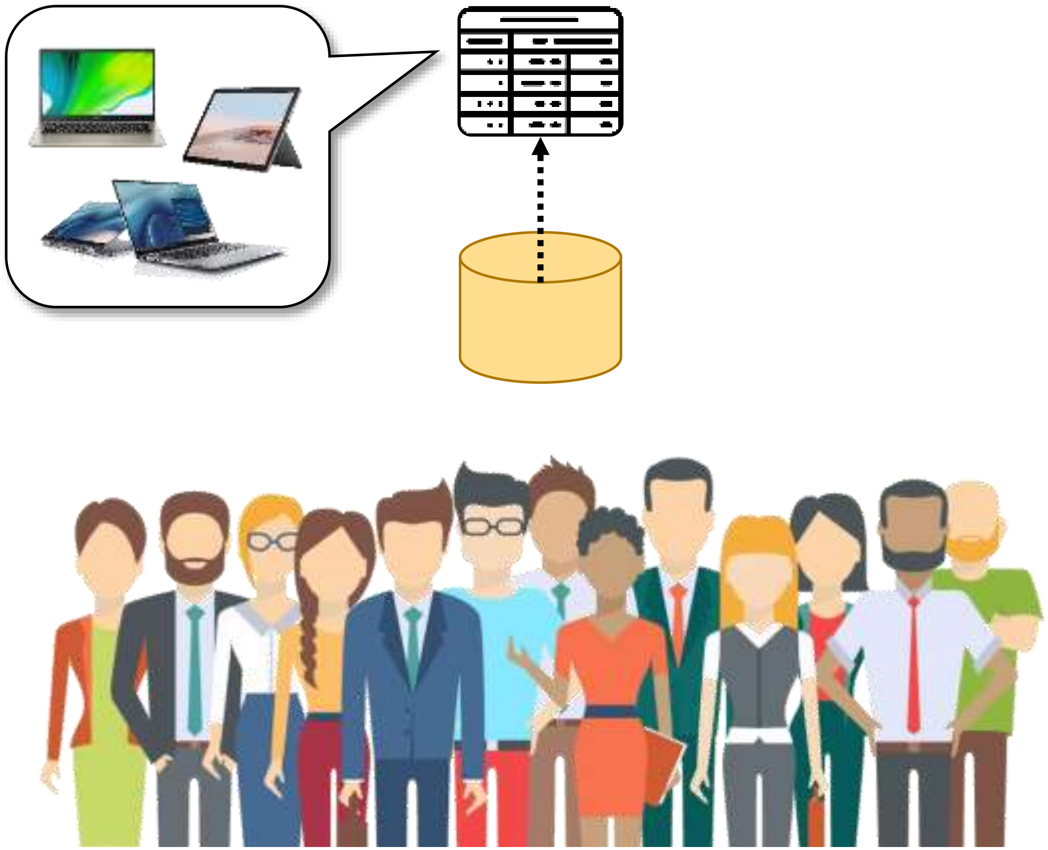
Bar-Ilan University

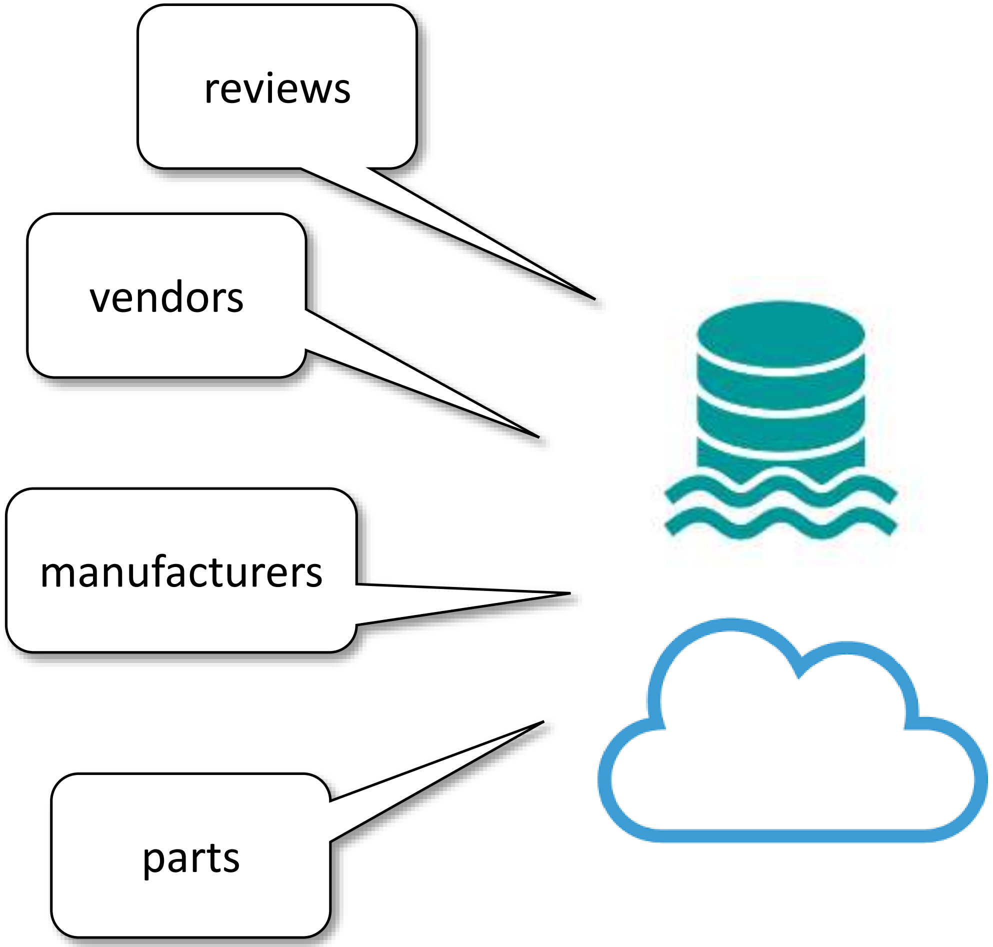
Automated Selection of Multiple Datasets for Extension by Integration

Yael Amsterdamer

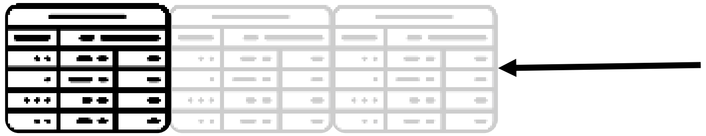
Moran Ben-Yehuda

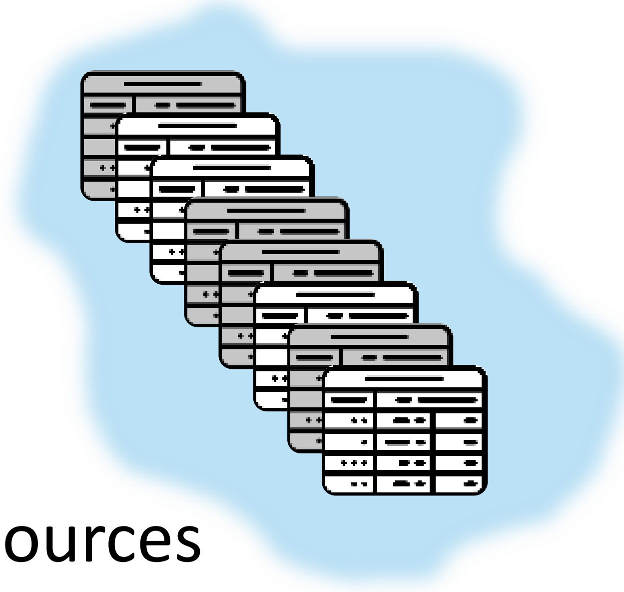
Extension by Integration





Scenario





- We have an initial data table (e.g., csv)
- We want to extend this table by integration with other sources
- Which ones to choose?
  - Amount of added data
  - Introduced errors
  - Completeness
  - Quality of matching to the initial table
- A greater challenge: integrating multiple tables

Candidates for integration

Initial dataset

Products

Prod	Manu.	Country
GreatPad X4000	BCnD	South Africa
GreatPad Y6000	BCnD	South Africa
Superb Vital	Macron	NULL
Smarterbook Elite	Netter	Saudi Arabia
Smarterbook Emerge	Netter	Saudi Arabia

CA (Companies in Africa)

Company	Located	Category	Rev.
BCnD	NULL	Technology	115.8
Macron	NULL	IT	155.3
Transact	Senegal	Finance	87.6
XYnZ	Tunisia	NULL	252.2

MEIT (Middle Eastern IT)

Name	Country	Revenue
Macron	Egypt	155
Netter	UAE	32
Opportune	Qatar	79
Promot	Israel	35
QueenTech	Jordan	28

Integration Result: Products & CA

Prod	Manu.	Country	Manu. Category	Manu. Rev.
GreatPad X4000	BCnD	South Africa	Technology	115.8
GreatPad Y6000	BCnD	South Africa	Technology	115.8
Superb Vital	Macron	NULL	IT	155.3
Smarterbook Elite	Netter	Saudi Arabia	NULL	NULL
Smarterbook Emerge	Netter	Saudi Arabia	NULL	NULL

Initial dataset

Candidates for integration

Products

Prod	Manu.	Country
GreatPad X4000	BCnD	South Africa
GreatPad Y6000	BCnD	South Africa
Superb Vital	Macron	NULL
Smarterbook Elite	Netter	Saudi Arabia
Smarterbook Emerge	Netter	Saudi Arabia

CA (Companies in Africa)

Company	Located	Category	Rev.
BCnD	NULL	Technology	115.8
Macron	NULL	IT	155.3
Transact	Senegal	Finance	87.6
XYnZ	Tunisia	NULL	252.2

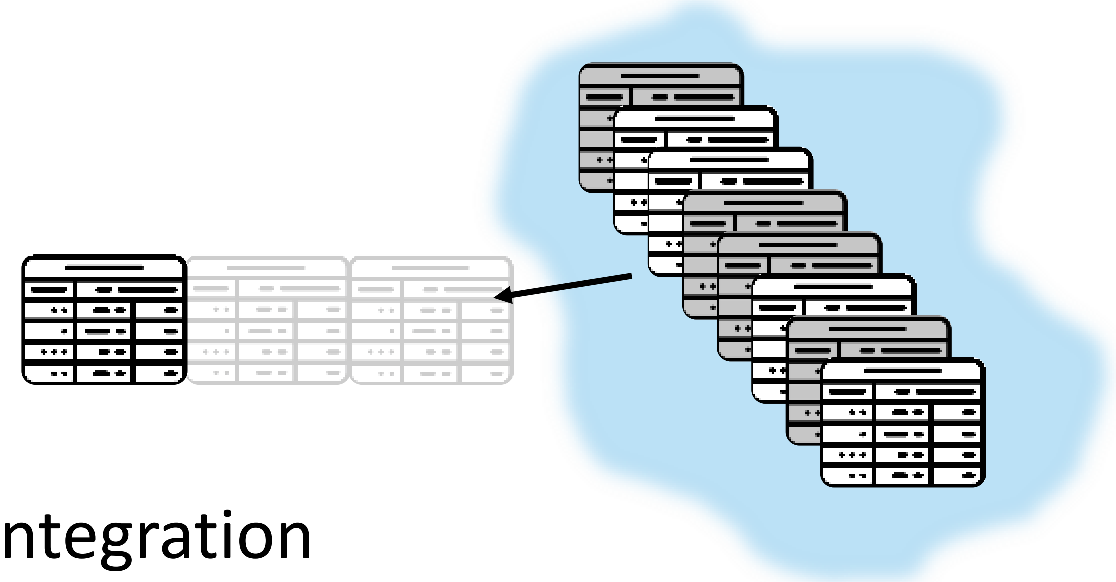
MEIT (Middle Eastern IT)

Name	Country	Revenue
Macron	Egypt	155
Netter	UAE	32
Opportune	Qatar	79
Promot	Israel	35
QueenTech	Jordan	28

Integration Result: Products & CA & MEIT

Prod	Manu.	Country	Manu. Category	Manu. Rev.
GreatPad X4000	BCnD	South Africa	Technology	115.8
GreatPad Y6000	BCnD	South Africa	Technology	115.8
Superb Vital	Macron	Egypt	IT	155.3
Smarterbook Elite	Netter	Saudi Arabia	NULL	32
Smarterbook Emerge	Netter	Saudi Arabia	NULL	32

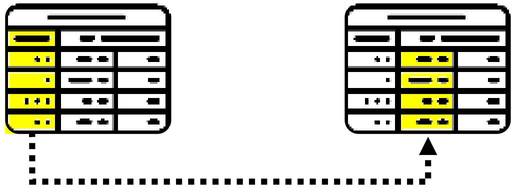
Our two main problems [CIKM 2021]:



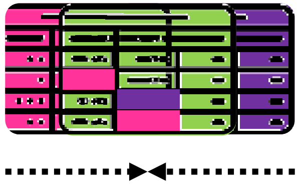
1. Define a metric for the "value" of integration

2. Efficiently find the subset of relations that maximizes it

Previous Work: Source Selection

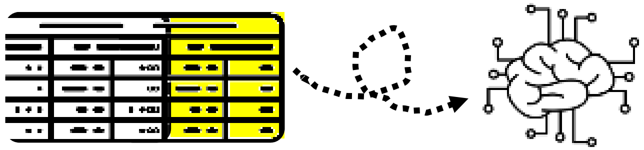


Finding links between relations
Domain Search
Finding joinable/unionable relations



Source Selection for Data Fusion

Metrics for table join in interactive data science (Zhang et al. 2020)



Data augmentation for machine learning (Chepurko et al. 2020)

Outline

- Problem definition
  - based on integration gain and cost
- Algorithms
- Experimental study

Metrics for Valuable Integration

- By properties of the integration result

  Some derivable from
  → Properties of integrated relations
  → Quality of integration

- Cost and gain of the integration

Let us start from the end:

- Assume a black-box for multi-relation integration
- **Integration gain** number of correct values in the integration result
  - Expected
  - How do we compute correctness likelihood?
- Integration cost:
  - **Incompleteness cost** number of NULLs
  - **Error cost** expected number of erroneous values
  - **Fixed cost per integration**

If the black box provides cell correctness probability estimation – we are done.

Properties of Integrated Relations

- Initial relation R_0, set of candidate relations $\mathcal{R} = \{R_1, R_2, \dots\}$
- Each R_i has
  - $U^i = \{U_1^i, U_2^i, \dots\}$ - attributes
  - $\mathrm{key}(R_i)$
  - Tuples with values (possibly NULL)
  - $\mathrm{P}^{\mathrm{correct}}(R_i)$ - probability of error in each value

Why compute this for the input and not the output?

Properties of Integration

- Many existing tools for data integration
  - Matching attributes
  - linking records
  - Mostly for relation pairs

Assume two black-box components:

(pairwise) attribute matching +indication for matching likelihood

(pairwise) record linking +indication for key match likelihood

- The integration result is

$$R = \mathcal{J}_{\text{res}}\left(\dots \mathcal{J}_{\text{res}}\left(\mathcal{J}_{\text{res}}\left(R_0, R_{i_1}\right), R_{i_2}\right) \dots, R_{i_m}\right)$$

Correctness Derivation: Linking tuples

- Let t be a tuple matched to tuple t' based on their values in U, U'
  - E.g., GreatPad X4000 matched to BCnD based on their values on attributes Manu. and Company
- Link weight for tuple t:

$$\mathrm{W}^{\mathrm{link}}(t) \coloneqq \mathrm{P}^{\mathrm{attMatch}}(U) \cdot \mathrm{W}^{\mathrm{correct}}(U, t) \cdot \mathrm{W}^{\mathrm{correct}}(U', t') \cdot \mathrm{P}^{\mathrm{valMatch}}\big(t(U), t'(U')\big)$$

Attributes are matched correctly

…and value in left table is correct

…and value in right table is correct

…and values indeed match

Products

Prod	Manu.	Country
GreatPad X4000	BCnD	South Africa
GreatPad Y6000	BCnD	South Africa
Superb Vital	Macron	NULL
Smarterbook Elite	Netter	Saudi Arabia
Smarterbook Emerge	Netter	Saudi Arabia

CA (Companies in Africa)

Company	Located	Category	Rev.
BCnD	NULL	Technology	115.8
Macron	NULL	IT	155.3
Transact	Senegal	Finance	87.6
XYnZ	Tunisia	NULL	252.2

MEIT (Middle Eastern IT)

Name	Country	Revenue
Macron	Egypt	155
Netter	UAE	32
Opportune	Qatar	79
Promot	Israel	35
QueenTech	Jordan	28

Integration Result: Products & CA & MEIT

Prod	Manu.	Country	Manu. Category	Manu. Rev.
GreatPad X4000	BCnD	South Africa	Technology	115.8
GreatPad Y6000	BCnD	South Africa	Technology	115.8
Superb Vital	Macron	Egypt	IT	155.3
Smarterbook Elite	Netter	Saudi Arabia	NULL	32
Smarterbook Emerge	Netter	Saudi Arabia	NULL	32

Correctness Derivation: Values

- Correctness of $t(U)$ in R
  - If R is an input relation, $\mathrm{W}^{\mathrm{correct}}(U, t) \coloneqq \mathrm{P}^{\mathrm{correct}}(R_i)$
  - If $t(U)$ was added in a new column as a result of integrating R',
    $$\mathrm{W}^{\mathrm{correct}}(U, t) \coloneqq \mathrm{W}^{\mathrm{link}}(t) \cdot \mathrm{W}^{\mathrm{correct}}(U, t')$$
  - If $t(U)$ was a NULL resolved by integrating R',
    $$\mathrm{W}^{\mathrm{correct}}(U, t) \coloneqq \mathrm{W}^{\mathrm{link}}(t) \cdot \mathrm{P}^{\mathrm{attMatch}}(U, U') \cdot \mathrm{W}^{\mathrm{correct}}(U', t')$$

Products

Prod	Manu.	Country
GreatPad X4000	BCnD	South Africa
GreatPad Y6000	BCnD	South Africa
Superb Vital	Mac	
Smarterbook Elite	Nett	
Smarterbook Emerge	Nett	

CA (Companies in Africa)

Company	Located	Category	Rev.
BCnD	NULL	Technology	115.8
Macron	NULL	IT	155.3
nsact	Senegal	Finance	87
Z	Tunisia	NULL	25

MEIT (Middle Eastern IT)

Name	Country	Revenue
Macron	Egypt	155
Netter	UAE	32
		79
		35
	n	28

A value in input relation

Resolved NULL; correctness depends on link and column matching

A value in a newly added column; correctness depends on link

Integratio sult: Products & CA

Prod	Manu.	Coun	Manu. Category	Manu. Rev.
GreatPad X4000	BCnD	Sou Africa	Technology	115.8
GreatPad Y6000	BCnD	So th Africa	Technology	115.8
Superb Vital	Macron	Egypt	IT	155.3
Smarterbook Elite	Netter	Saudi Arabia	NULL	32
Smarterbook Emerge	Netter	Saudi Arabia	NULL	32

Formal Pairwise Definitions

Gain expected number of correct values.

$$\text{gain}(R_i, R_j) := \sum_{t \in \mathcal{I}_{\text{res}}(R_i,R_j)} \sum_{\substack{U \in U_{i,j} \\ t(U) \neq \text{NULL}}} \text{W}^{\text{correct}}(U, t)$$

Integration cost:

- **Incompleteness cost** number of NULL in the integration result

$$\text{Cost}_{\text{NULL}}(\mathcal{I}_{\text{res}}(R_i, R_j))$$

- **Error cost** expected number of erroneous values.

$$\text{Cost}_{\text{err}}(\mathcal{I}_{\text{res}}(R_i, R_j)) := \sum_{t \in \mathcal{I}_{\text{res}}(R_i,R_j)} \sum_{\substack{U \in U_{i,j} \\ t(U) \neq \text{NULL}}} (1 - \text{W}^{\text{correct}}(U, t))$$

- **Fixed cost per integration** e.g., monetary cost

$$\text{Cost}_{\text{fixed}}(\mathcal{I}_{\text{res}}(R_i, R_j)) := \text{Cost}_{\text{fixed}}(R_i) + \text{Cost}_{\text{fixed}}(R_j)$$

OPT-EXTENSION

Find Sub-sequence $R_{i_1}, R_{i_2}, \ldots, R_{i_m}$

Integration Result $R = \mathcal{I}_{\mathrm{res}}\left(\ldots \mathcal{I}_{\mathrm{res}}\left(\mathcal{I}_{\mathrm{res}}\left(R_0, R_{i_1}\right), R_{i_2}\right) \ldots, R_{i_m}\right)$

Maximize metric

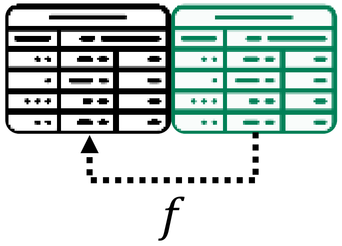
$$\mathrm{score}(R, \alpha, \beta, \gamma) \coloneqq \mathrm{gain}(R) - (\alpha\, \mathrm{Cost}_{\mathrm{NULL}}(R) + \beta\, \mathrm{Cost}_{\mathrm{err}}(R) + \gamma\, \mathrm{Cost}_{\mathrm{fixed}}(R))$$

Hardness of OPT-EXTENSION

- OPT-EXTENSION is $\mathrm{FP}^{\mathrm{NP}}$*-hard*
    - By a reduction from SET COVER
    - Membership result
- Score function is not monotone / convex

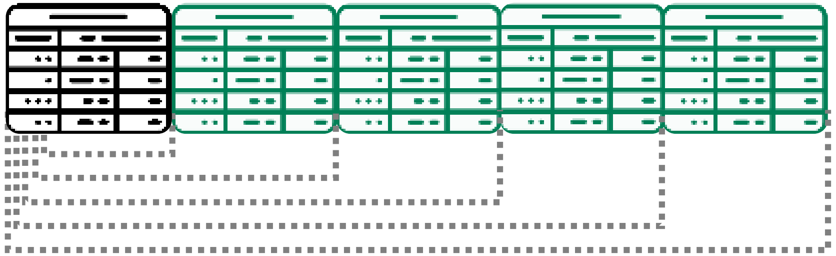
Algorithms

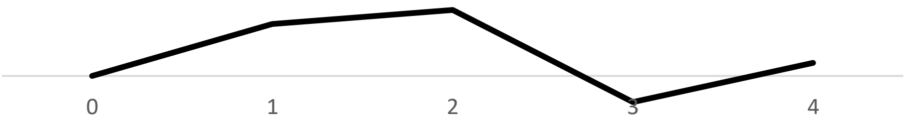
Our Solution Scheme



- Iteratively select the next relation to integrate
  - using function f
- Exhaustively integrate
- Select intermediate best result

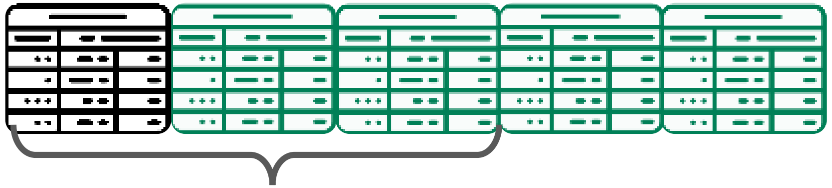
Our Solution Scheme

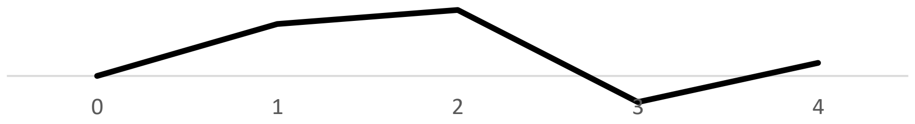




- Iteratively select the next relation to integrate
  - using function f
- Exhaustively integrate
- Select intermediate best result

Our Solution Scheme





- Iteratively select the next relation to integrate
  - using function f
- Exhaustively integrate
- Select intermediate best result

Selection Criteria

EDMINT-Greedy:

- Greedily maximize the score at each iteration
- Empirically achieves near-optimal scores
- But: performs many integrations

Selection Criteria

EDMINT-Opt:

- Reduce number of integrations by
  - Identifying relations that cannot increase the score
  - Identifying relations whose marginal contribution is fixed

---

Algorithm 2: `Edmint-Opt` implementation of f (selection of the next relation to integrate in Algorithm 1)

Input: R: initial relation; $\mathcal{R} = \{R_1, \ldots, R_n\}$: set of candidate relations; $\mathcal{I}$: integration black-box; α, β, γ: weights for score function

Output: $R_{\max} \in \mathcal{R}$: single relation to integrate; $\mathcal{R}$ updated set of candidates

for $R_i \in \mathcal{R}$ **do**
    maxDelta $\leftarrow$ gain$(R_i) - \beta\,\mathrm{Cost_{err}}(R_i) - \gamma\,\mathrm{Cost_{fixed}}(R_i)$;
    if maxDelta < 0 **then** $\mathcal{R} \leftarrow \mathcal{R} - \{R_i\}$;
$R_{\max} \leftarrow \emptyset$;
maxScore $\leftarrow -\infty$;
for $R_i \in \mathcal{R}$ **do**
    if $\forall U_k^i \in U^i$, $\mathrm{attMatch}^+_{\mathrm{R},\mathcal{R}}(U_k^i) = \emptyset$ **then**
        delta $\leftarrow$ score$(\mathcal{I}_{\mathrm{res}}(R, R_i))$ $-$ score(R);
        if $\mathrm{attMatch}^{-1}_{\mathrm{R,R_i}}(\mathrm{key}(R_i)) \neq \perp$ *and* delta > 0 **then**
         Return $R_i, \mathcal{R}$;
        else $\mathcal{R} \leftarrow \mathcal{R} - \{R_i\}$;
    else if $\mathrm{attMatch}^{-1}_{\mathrm{R,R_i}}(\mathrm{key}(R_i)) \neq \perp$ **then**
        $R' \leftarrow \mathcal{I}_{\mathrm{res}}(R, R_i)$;
        Update $R_{\max}$, maxScore if the score of R' is greater;
return $R_{\max}, \mathcal{R}$;

Selection Criteria

EDMINT-Opt:

- Reduce number of integrations by
  - Identifying relations that cannot increase the score
  - Identifying relations whose marginal contribution is fixed

Algo... f (selection
of th... 1)

Inp... set of candidate
... β, γ: weights

Output: R... single relation to integrate; $\mathcal{R}$ updated
... candidates

for R_i ... **do**
  maxDelta $\leftarrow$ gain$(R_i) - \beta\,\mathrm{Cost_{err}}(R_i) - \gamma\,\mathrm{Cost_{fixed}}(R_i)$;
  if maxDelta < 0 **then** $\mathcal{R} \leftarrow \mathcal{R} - \{R_i\}$;
$R_{\max} \leftarrow \emptyset$;
maxScore $\leftarrow -\infty$;
for $R_i \in \mathcal{R}$ **do**
  if $\forall U_k^i \in U^i$, $\mathrm{attMatch}^{+}_{R,\mathcal{R}}(U_k^i) = \emptyset$ **then**
    delta $\leftarrow$ score$(\mathcal{I}_{\mathrm{res}}(R, R_i))$ − score(R);
    if $\mathrm{attMatch}^{-1}_{R,R_i}(\mathrm{key}(R_i)) \neq \perp$ *and* delta > 0 **then**
      Return $R_i, \mathcal{R}$;
    else $\mathcal{R} \leftarrow \mathcal{R} - \{R_i\}$;
  else if $\mathrm{attMatch}^{-1}_{R,R_i}(\mathrm{key}(R_i)) \neq \perp$ **then**
    $R' \leftarrow \mathcal{I}_{\mathrm{res}}(R, R_i)$;
    Update $R_{\max}$, maxScore if the score of R' is greater;
return $R_{\max}, \mathcal{R}$;

The maximal marginal contribution of a relation (if all non-NULL cells are added, and there are no added NULLs)

There are no additional matches of this candidate relation to other candidates

Integrations are still a bottleneck

- We use an implementation based on locality sensitive hashing (LSH):
  - Attribute sketches used to estimate matching probability
  - An index used to find matches in constant time

  - Depends on the attribute matching method

Experimental Study

Compared algorithms

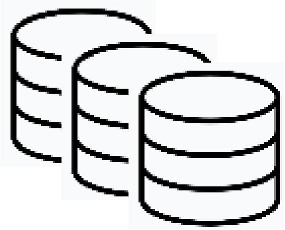
- AccDesc - f greedily selects the most accurate relation that can be integrated
- Random - f selects a random relation that can be integrated
- Brute-Force
- EDMINT-Greedy
- EDMINT-Opt

Metrics

We consider three general types of metrics for the integration result:

- Score of the integration
- Number of rounds
- Number of integrations

Datasets

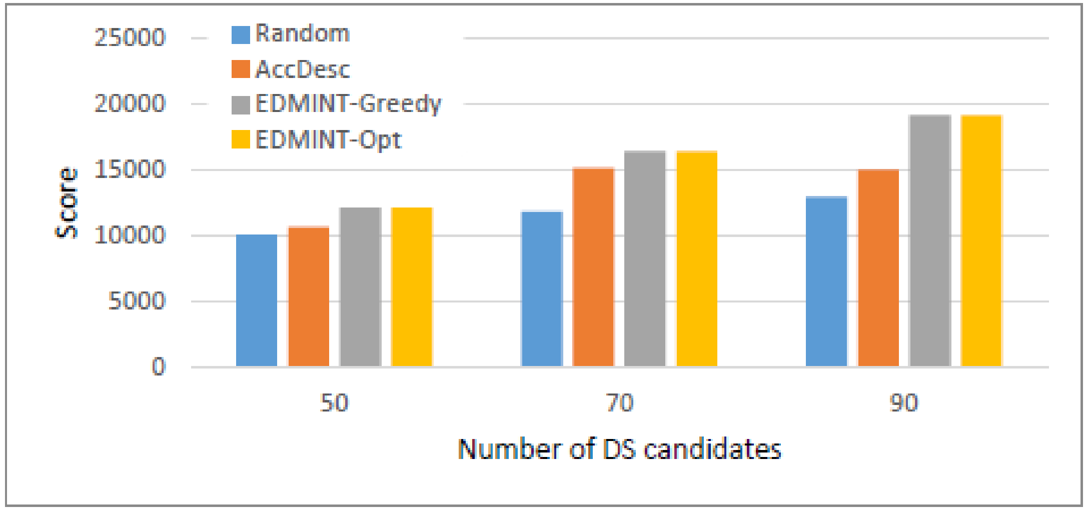


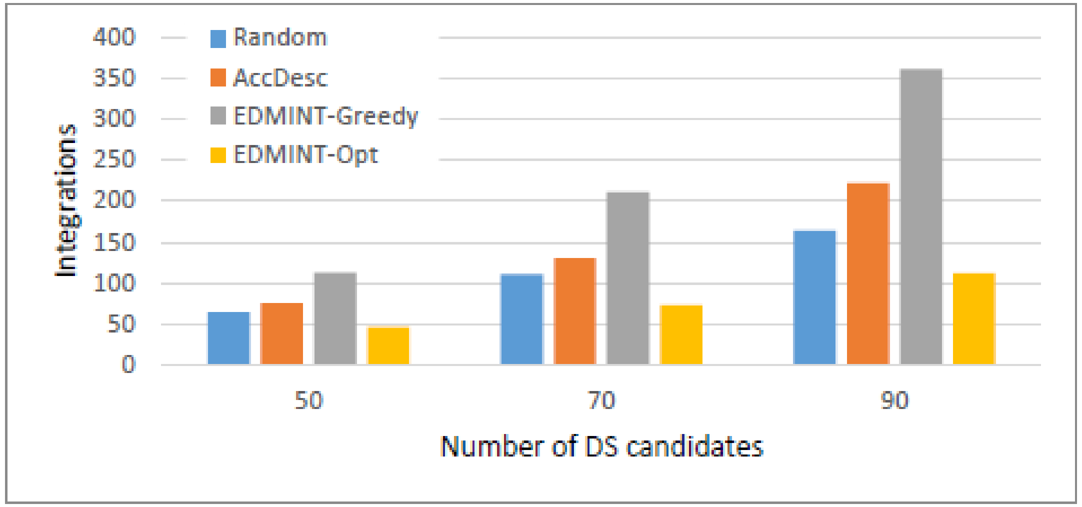
- **Kaggle Collection**: 40 relations related to movies and books
  → scenario: user already collected relevant datasets
- **Medley Collection.** 100 relations on various topics
  → scenario: using a data lake

- Each relation consists of 120-1M tuples and 2-67 attributes.

Experimental Results

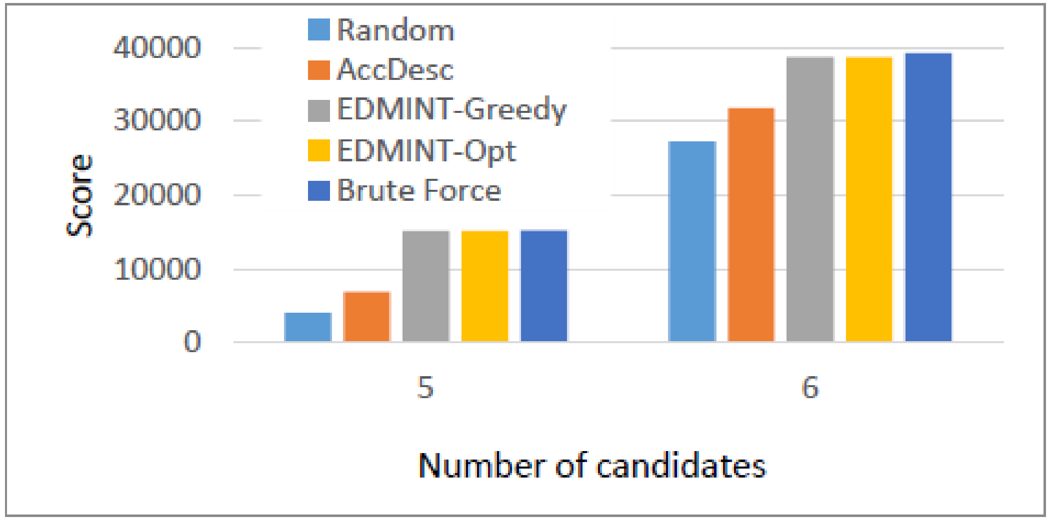
Varying candidate collection sizes:

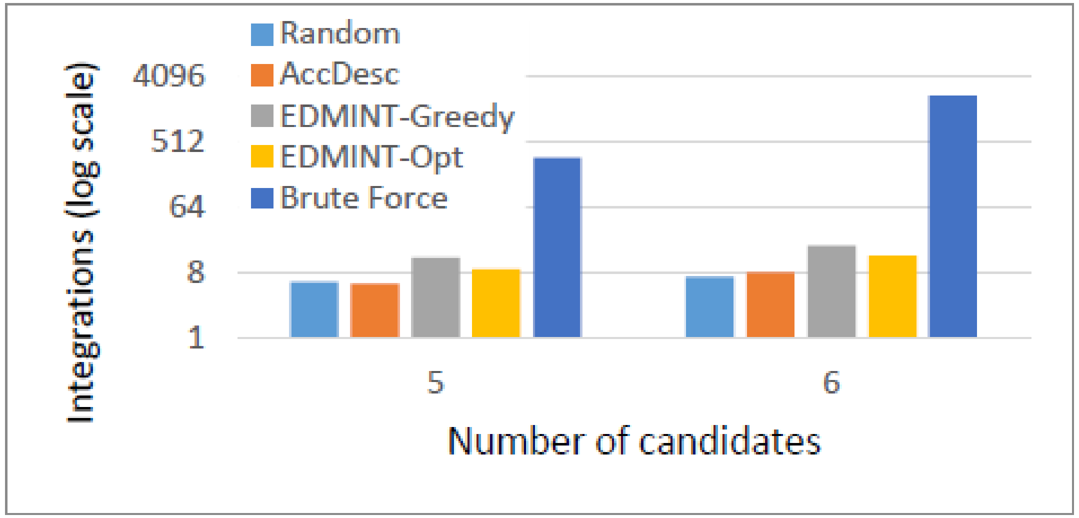




Experimental Results

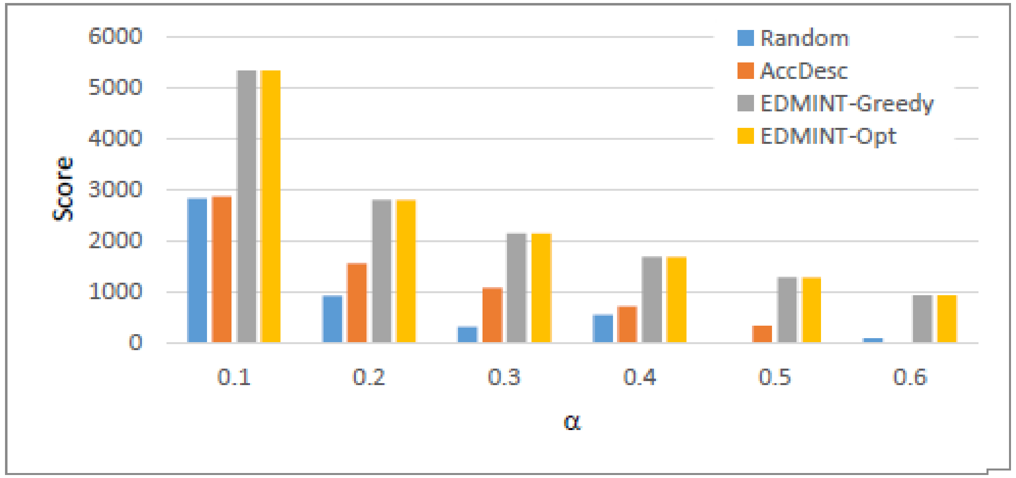
Comparison to the optimal algorithm:

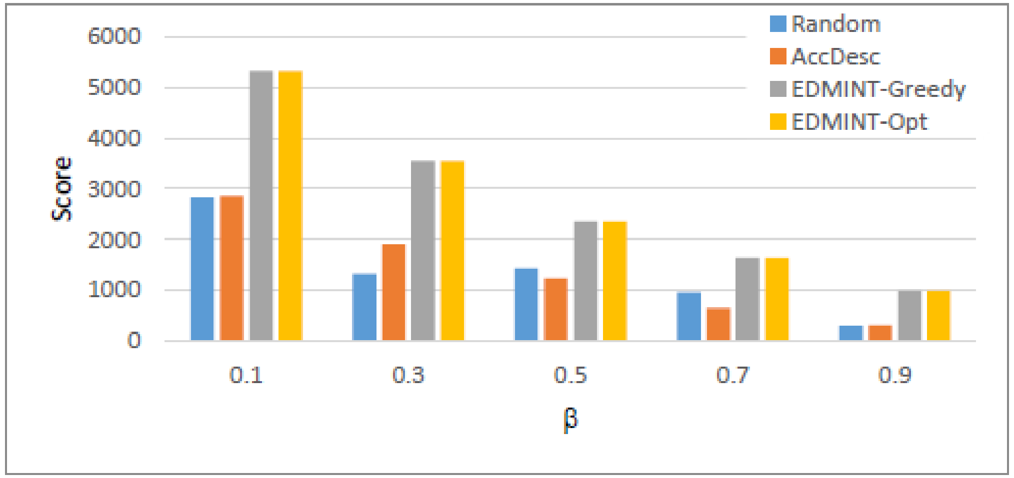




Experimental Results

Varying the parameters:

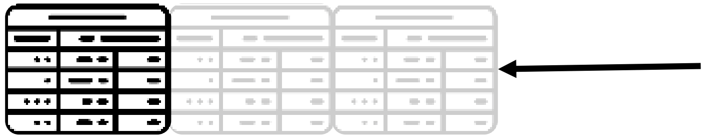


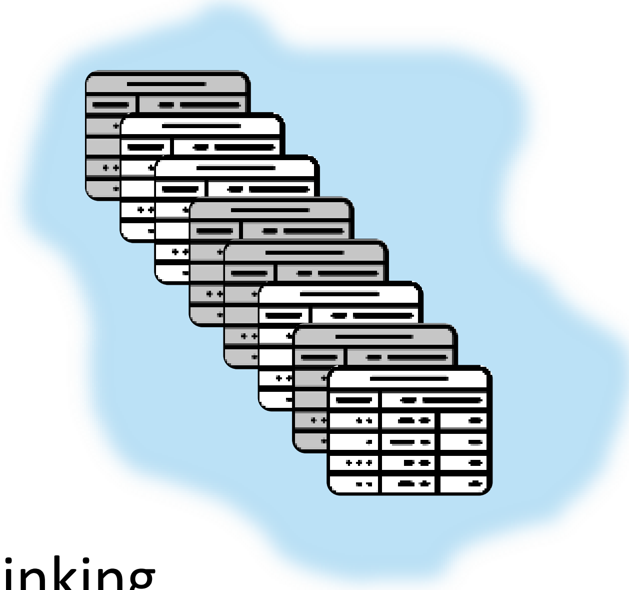


Execution times

$\lvert\mathcal{R}\rvert$	$\sum \lvert U^i \rvert$	% match	Index time	Avg. Integration time			
				Random	AccDesc	`Edmint-Greedy`	`Edmint-Opt`
90	3628	0.11	07:32	<00:01	<00:01	<00:01	<00:01
70	2221	0.15	05:09	<00:01	<00:01	<00:01	<00:01
50	2358	0.13	03:33	<00:01	<00:01	<00:01	<00:01
19	300	0.63	03:07	00:02	00:01	00:02	00:02
12	183	0.85	00:35	00:01	00:01	<00:01	<00:01
5	76	0.4	00:22	<00:01	<00:01	<00:01	00:01

Summary





- We defined the problem of extension by integration
  - Cost and gain of integration
  - Using pair-wise black-boxes for attribute matching and tuple linking
  - Direct optimization is hard
- We proposed a scheme and algorithms for the solution
- Experiments on real data
  - Near-optimal score
  - Efficiency by reducing the number of integrations
- Our solution can be combined with various integration methods

Future work

- Extending non-relational data
- Accounting explicitly for other aspects of integration
  - Relevance
  - Data cleaning
  - Data fusion
- Perform automated transformations (group by, filter, pivot) on relations to improve integration quality

Thank You!