ASP-Based System to Collective Entity Resolution

Zhiliang (Leon) Xiang, Meghyn Bienvenu, Gianluca Cima,
Víctor Gutiérrez Basulto, Yazmín Ibáñez-García

Cardiff Knowledge Representation & Reasoning (KRR) Group

the LaBRI Research Lab

Sapienza University of Rome

Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	123456	brufen
p2	John Smith	32	12345-6	aspirin

Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	123456	brufen
p2	**John Smith**	32	12345-6	aspirin

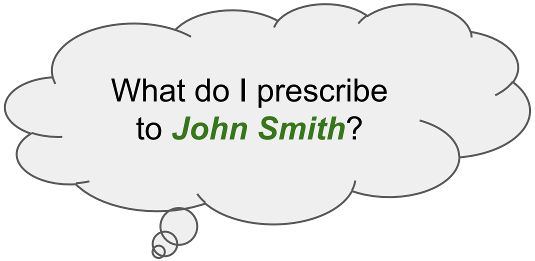
Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	123456	brufen
p2	**John Smith**	32	12345-6	aspirin



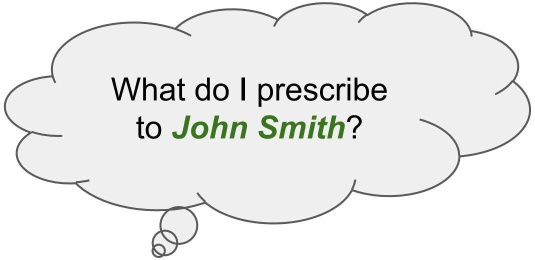
Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Patient

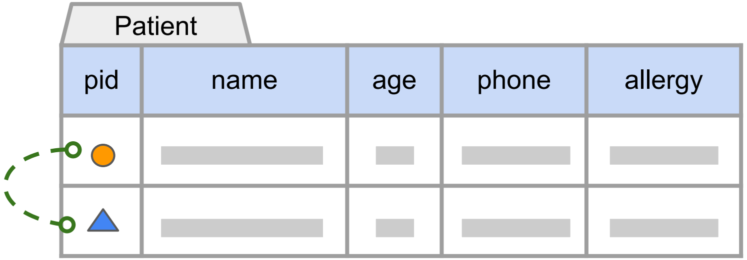
pid	name	age	phone	allergy
p1	J.Smith	32	123456	brufen
p2	**John Smith**	32	12345-6	aspirin



Critical to **data quality** and **decision making**!

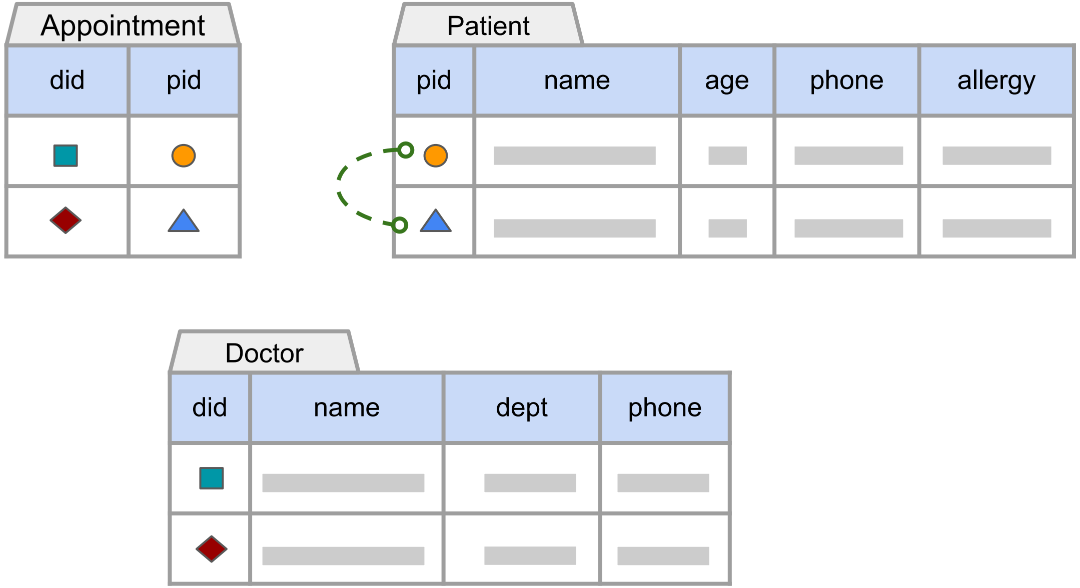
Entity Resolution (ER)

Traditional: within **single-table** or **table pair** (same entity type)



Entity Resolution (ER)

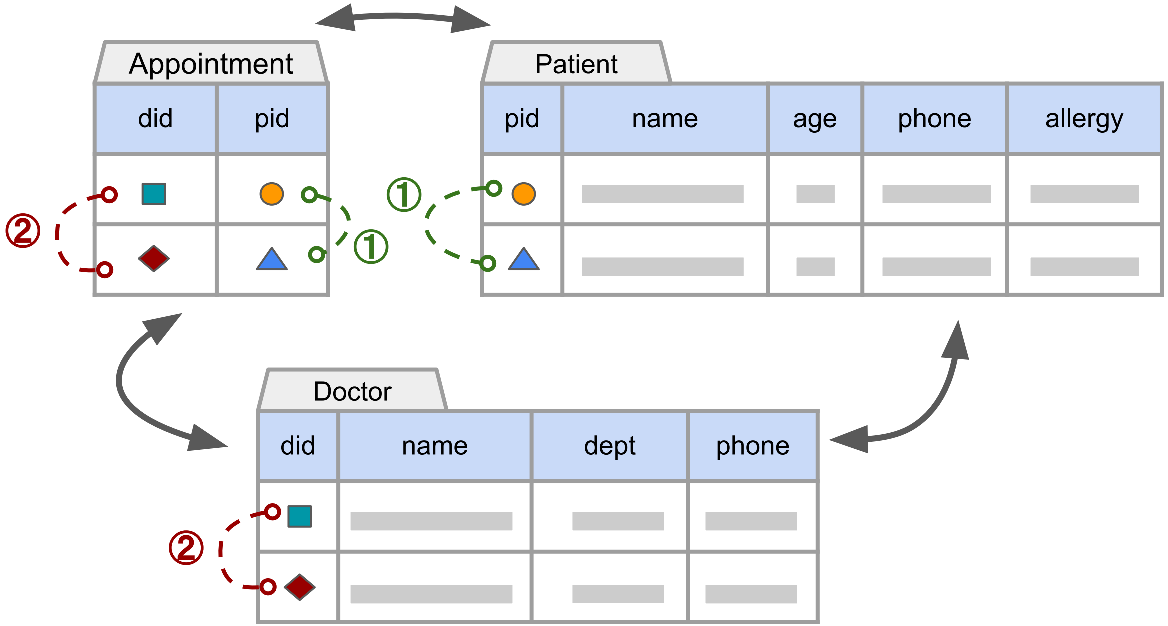
Traditional: within **single-table** or **table pair** (same entity type)



Entity Resolution (ER)

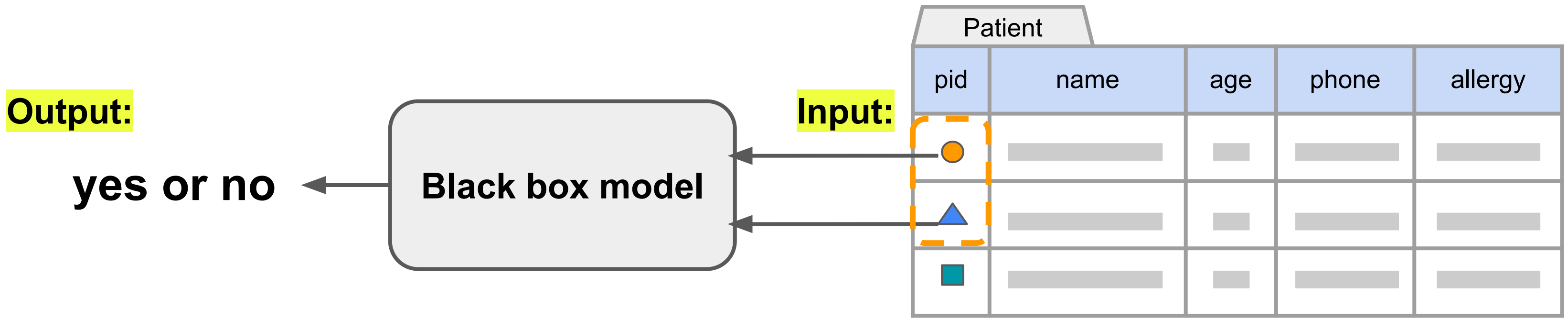
Traditional: within **single-table** or **table pair** (same entity type)

Collective: across **multiple tables**, exploit **inter-dependencies** between tables



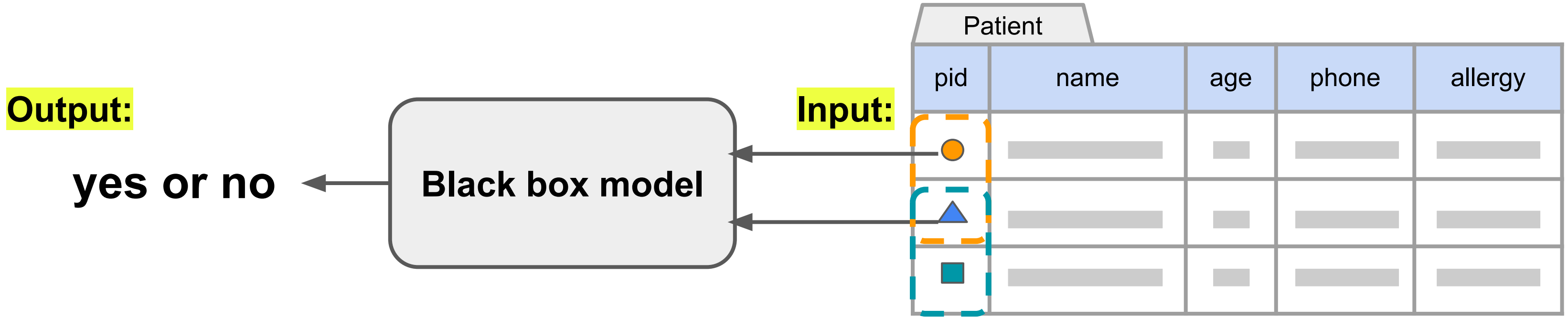
Existing work

Deep learning methods excel in **pairwise** setting,



Existing work

Deep learning methods excel in **pairwise** setting,



but …

- Predictions may be **inconsistent**, e.g. ● = ▲ & ▲ = ■ but ■ ≠ ●
- Not clear how to support **collective ER**
- **Explanation** matters

Existing work

Novel rule-based framework:

A Logical Approach to Collective Entity Resolution (LACE) *[Bienvenu et al. PODS 2022, KR 2023]*

- **declarative:** based on logic
- **collective:** capture complex **interdependencies** between tables
- **explainable**: "**why** are *J. Smith* and *John Smith* deemed to be the **same** patient?"

Motivation

- A real-life implementation of LACE is **absent**
- No existing system supports both **global** **&** **contextual** resolutions
- **No or only one optimality** for ER solution, alternatives?

Motivation

- A real-life implementation of LACE is **absent**

Critical to **feasibility** and **usefulness** of the formalism!

- No existing system supports both **global** **&** **contextual** resolutions

Very important to **correctness** and **completeness**!

- **No or only one optimality** for ER solution, alternatives?

Offering **flexibility** to tailor **preferred** ER solutions!

Motivation

Goal: To develop **ASP-based** implementation of LACE

Why ASP?

- Proven correspondence
- Many functionalities available for use, e.g. preference, explanation
- *"ASP = DB+LP+KR+SAT"*, testing on DB reasoning problems in the wild

Formalism

ER rule

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

$$Patient(i, x, \mathbf{n}, \mathbf{a}, \mathbf{t}, l) \wedge Patient(i', y, \mathbf{n'}, \mathbf{a}, \mathbf{t}, l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow \mathbf{EqO(x,y)}$$

ER rule

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

$$Patient(i, x, \mathbf{n}, \mathbf{a}, \mathbf{t}, l) \wedge Patient(i', y, \mathbf{n'}, \mathbf{a}, \mathbf{t}, l') \wedge \mathbf{n} \approx \mathbf{n'} => \mathbf{EqO(x,y)}$$

ER rule

Database D

Appointment	
did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient				
pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor			
did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

$$Patient(i, x, \mathbf{n}, \mathbf{a}, \mathbf{t}, l) \wedge Patient(i', y, \mathbf{n'}, \mathbf{a}, \mathbf{t}, l') \wedge \mathbf{n \approx n'} \Rightarrow \mathbf{EqO(x,y)}$$

⇩

pid of two patients are the same if they have **similar *name***, and **same *age* and *phone* number**

LACE Specification

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow \mathbf{EqO(x,y)}$

Soft rule: $Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow \mathbf{EqO(x,}$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \perp$

LACE Specification

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow \mathbf{EqO(x,y)}$

Soft rule: $Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p})$ $\sim>$ $\mathbf{EqO(x,}$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \perp$

LACE Specification

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(i, x, \mathbf{n}, \mathbf{a}, \mathbf{t}, l) \wedge Patient(i', y, \mathbf{n'}, \mathbf{a}, \mathbf{t}, l') \wedge \mathbf{n \approx n'} \Rightarrow \mathbf{EqO(x,y)}$

Soft rule: $Doctor(i, \mathbf{x}, n, \mathbf{d}, \mathbf{t}) \wedge Doctor(i', \mathbf{y}, n', \mathbf{d}, \mathbf{t}) \wedge App(i'', \mathbf{x}, \mathbf{p}) \wedge App(i''', \mathbf{y}, \mathbf{p}) \rightsquigarrow \mathbf{EqO(x,}$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \perp$

Object and Value

Database D

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

object

data value

Specification L

$$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} \Rightarrow \mathbf{EqO(x,y)}$$

$$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow \mathbf{EqO(x,y)}$$

$$Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \bot$$

Object and Value

Database D

Patient					
tid	1:pid	2:name	3:age	4:phone	5:allergy
t1	p1	J.Smith	32	12345	brupen
t2	p2	John Smith	32	12345	aspirin
t3	p3	Jerry Smith	32	1234-5	ibuprofen
t4	p4	J.Smith	32	6789	ibuprofen

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} => \mathbf{EqO(x,y)}$ object rule

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \sim> \mathbf{EqO(x,y)}$

$Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n \neq n'} => \perp$

$Patient(i, \mathbf{p},\mathbf{n},a,\mathbf{t},l) \wedge Patient(i', \mathbf{p},\mathbf{n'},a',\mathbf{t},l') => \mathbf{EqV(<i,2>,<i',2>)}$ value rule

$\mathbf{EqO/2}$ stores pairs of constants

$\mathbf{EqV/2}$ stores pairs of "locations"

Object and Value

Database D

Patient					
tid	1:pid	2:name	3:age	4:phone	5:allergy
t1	p1	J.Smith	32	12345	brupen
t2	p2	John Smith	32	12345	aspirin
t3	p3	Jerry Smith	32	1234-5	ibuprofen
t4	p4	J.Smith	32	6789	ibuprofen

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} \Rightarrow \mathbf{EqO(x,y)}$ **object rule**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow \mathbf{EqO(x,y)}$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \perp$

$Patient(i, \mathbf{p},\mathbf{n},a,\mathbf{t},l) \wedge Patient(i', \mathbf{p},\mathbf{n'},a',\mathbf{t},l') \Rightarrow \mathbf{EqV(\langle i,2\rangle,\langle i',2\rangle)}$ **value rule**

EqRels E,V

$E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $V = \{(\langle t_1,2\rangle,\langle t_2,2\rangle),(\langle t_3,2\rangle,\langle t_4,2\rangle)\}$

Object and Value

Database D

Patient					
tid	1:pid	2:name	3:age	4:phone	5:allergy
t1	p1	J.Smith	32	12345	brupen
t2	p2	John Smith	32	12345	aspirin
t3	p3	Jerry Smith	32	1234-5	ibuprofen
t4	p4	J.Smith	32	6789	ibuprofen

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow \mathbf{EqO(x,y)}$ **object rule**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow \mathbf{EqO(x,y)}$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \perp$

$Patient(i, \mathbf{p},\mathbf{n},a,\mathbf{t},l) \wedge Patient(i', \mathbf{p},\mathbf{n'},a',\mathbf{t},l') \Rightarrow \mathbf{EqV(<i,2>,<i',2>)}$ **value rule**

EqRels $\boldsymbol{E},\boldsymbol{V}$

$\boldsymbol{E} = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\boldsymbol{V} = \{(<t_1,2>,<t_2,2>),(<t_3,2>,<t_4,2>)\}$

Induced database

Induced database $D_{E,V}$

Patient					
tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} \Rightarrow \mathbf{EqO(x,y)}$ **object rule**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow \mathbf{EqO(x,y)}$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \perp$

$Patient(i, p,\mathbf{n},a,t,l) \wedge Patient(i', p,\mathbf{n'},a',t,l') \Rightarrow \mathbf{EqV(<i,2>,<i',2>)}$ **value rule**

EqRels E,V

$E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $V = \{(<t_1,2>,<t_2,2>),(<t_3,2>,<t_4,2>)\}$

replace each **object** constant c with the set $\{c' \mid (c,c') \in E\}$,

replace each **value** constant at the location $<t,i>$ with the set $\{t'[j] \mid (<t,i>,<t',j>) \in V\}$

Rule evaluation: equality

Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

- Same variable mapped to different sets depending on the context (tuple) it occurs
- Final assignment is the intersection of such sets

Query evaluation on $D_{E,V}$

$$\mathit{Patient}(i,\ \overset{1}{p},\overset{2}{n},\overset{3}{a},\overset{4}{t},\overset{5}{l}) \wedge \mathit{Patient}(i',\ \overset{1}{p},\overset{2}{n},\overset{3}{a'},\overset{4}{t'},\overset{5}{l'}) \wedge t \approx t' \wedge l \neq l'$$

$\{t1\}$ {J.Smith,John Smith} $\{t3\}$ {J.Smith,Jerry Smith}

{J.Smith} **satisfied**

Rule evaluation: equality

Induced database $D_{E,V}$

Patient					
tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

- Same variable mapped to different sets depending on the context (tuple) it occurs
- Final assignment is the intersection of such sets

Query evaluation on $D_{E,V}$

$$Patient(i, \overset{1}{p}, \overset{2}{n}, \overset{3}{a}, \overset{4}{t}, \overset{5}{l}) \wedge Patient(i', \overset{1}{p}, \overset{2}{n}, \overset{3}{a'}, \overset{4}{t'}, \overset{5}{l'}) \wedge t \approx t' \wedge l \neq l'$$

{t1} {p1,p2} {t3} {p3,p4}

$\varnothing$ **not satisfied**

Rule evaluation: inequality

Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

- Same variable mapped to different sets depending on the context (tuple) it occurs
- Final assignment is the intersection of such sets
- For $l \neq l'$, satisfied if intersection mapped sets of l and l' is empty
- For $t \approx t'$, satisfied if $\approx$ holds for two constants from the mapped sets of t and t', resp

Query evaluation on $D_{E,V}$

$$Patient(\overset{}{i}, \overset{1}{p}, \overset{2}{n}, \overset{3}{a}, \overset{4}{t}, \overset{5}{l}) \wedge Patient(i', \overset{1}{p}, \overset{2}{n}, \overset{3}{a'}, \overset{4}{t'}, \overset{5}{l'}) \wedge t \approx t' \wedge l \neq l'$$

$i \to \{t1\}$, $l \to$ {brupen}, $i' \to \{t3\}$, $l' \to$ {ibuprofen}, {brupen} ∩ {ibuprofen} $\to \varnothing$

satisfied

Rule evaluation: similarity

Induced database $D_{E,V}$

Patient					
tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

- Same variable mapped to different sets depending on the context (tuple) it occurs
- Final assignment is the intersection of such sets
- For $l \neq l'$, satisfied if intersection mapped sets of l and l' is empty
- For $t \approx t'$, satisfied if $\approx$ holds for two constants from the mapped sets of t and t', resp

Query evaluation on $D_{E,V}$

$$Patient(\underset{\{t1\}}{i}, \overset{1}{p}, \overset{2}{n}, \overset{3}{a}, \underset{\{12345\}}{\overset{4}{t}}, \overset{5}{l}) \wedge Patient(\underset{\{t3\}}{i'}, \overset{1}{p}, \overset{2}{n}, \overset{3}{a'}, \underset{\{1234\text{-}5\}}{\overset{4}{t'}}, \overset{5}{l'}) \wedge t \approx t' \wedge l \neq l'$$

12345 ≈ 1234-5

satisfied

Solution

Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} \Rightarrow \mathbf{EqO(x,y)}$ **object merge**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow \mathbf{EqO(x,y)}$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \bot$

$Patient(i, p,\mathbf{n},a,t,l) \wedge Patient(i', p,\mathbf{n'},a',t,l') \Rightarrow \mathbf{EqV(<i,2>,<i',2>)}$ **location merge**

EqRels E,V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(<t_1,2>,<t_2,2>),(<t_3,2>,<t_4,2>)\}$

A *solution* is an $<E,V>$ s.t. $D_{E,V}$ satisfies **H** and **C**

Solution

Induced database $D_{E,V}$

Patient					
tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

Specification L

$Patient(i, x, \mathbf{n}, \mathbf{a}, \mathbf{t}, l) \wedge Patient(i', y, \mathbf{n'}, \mathbf{a}, \mathbf{t}, l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow \mathbf{EqO(x,y)}$ **object merge**

$Doctor(i, \mathbf{x}, n, \mathbf{d}, \mathbf{t}) \wedge Doctor(i', \mathbf{y}, n', \mathbf{d}, \mathbf{t}) \wedge App(i'', \mathbf{x}, \mathbf{p}) \wedge App(i''', \mathbf{y}, \mathbf{p}) \rightsquigarrow \mathbf{EqO(x,y)}$

$Patient(i, p, n, a, t, l) \wedge Patient(i', p, n', a', t', l') \wedge \mathbf{n \neq n'} \Rightarrow \bot$

$Patient(i, p, \mathbf{n}, a, t, l) \wedge Patient(i', p, \mathbf{n'}, a', t, l') \rightsquigarrow \mathbf{EqV(\langle i,2\rangle, \langle i',2\rangle)}$ **location merge**

EqRels E,V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(\langle t_1,2\rangle,\langle t_2,2\rangle),(\langle t_3,2\rangle,\langle t_4,2\rangle)\}$

A *solution* is an $\langle E,V\rangle$ s.t. $D_{E,V}$ satisfies **H** and **C**

Space of solutions → Set inclusion optimal solutions

System

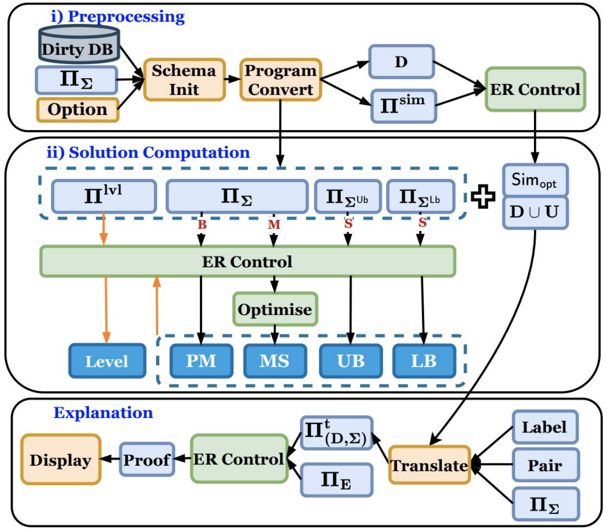
The ASPEn system *[Xiang et al. KR 2024,KR 2025]*

Input:

- Database, ASP Encoding
- Options specify desired output

Output:

- Merge lower/upper bound (LB/UB)
- Exact set of brave consequence (set optim)
- **Fixed # of optimal solutions**
- Graphical explanation of a possible merge (s)



facts

```
eqo(p3,p3).
eqv(t1,2,t1,2). sim(JohnSmith,JSmith,90).
…
loc(t1,2,JohnSmith). patient(t1,p1,JohnSmith,25,12345,brupen).
```

rules

```
eqo(X,Y) :- patient(I1,X,A1,T1,_),patient(I2,Y,A2,T2,_),val(I1,2,N),val(I2,2,N).
activeo(X,Y) :- doctor(I1,X,_,D,T),doctor(I2,Y,_,D,T),
               app(I3,X1,P1),app(I4,Y1,P2), eqo(X1,X),eqo(Y1,Y), eqo(P1,P2).
eqv(I1,2,I2,2):- patient(I',P2,_,_,_),eqo(P1,P2),patient(I,P1,_,_,_),
               val(I1,2,T1),val(I2,2,T2),sim(T1,T2,S),S>90.
```

axioms

```
val(I1,2,N2):- eqv(I1,2,I2,2),loc(I2,2,N2).
i(I1,2,I2,2):- val(I1,2,V),val(I2,2,V), not empty(V).
eqo(X,Y):- eqo(Y,X). eqo(X,Y):- eqo(X,Z),eqo(Z,Y).
{eqo(X,Y)}:- activeo(X,Y).
```

cons

```
:- patient(I1,P1,_,_,_),patient(I2,P2,_,_,_),eqo(P1,P2), not i(I1,2,I2,2).
```

facts

```
eqo(p3,p3).
eqv(t1,2,t1,2). sim(JohnSmith,JSmith,90).
…
loc(t1,2,JohnSmith). patient(t1,p1,JohnSmith,25,12345,brupen).
```

rules

```
eqo(X,Y) :- patient(I1,X,A1,T1,_),patient(I2,Y,A2,T2,_),val(I1,2,N),val(I2,2,N).
activeo(X,Y) :- doctor(I1,X,_,D,T),doctor(I2,Y,_,D,T),
               app(I3,X1,P1),app(I4,Y1,P2), eqo(X1,X),eqo(Y1,Y), eqo(P1,P2).
eqv(I1,2,I2,2):- patient(I',P2,_,_,_),eqo(P1,P2),patient(I,P1,_,_,_),
               val(I1,2,T1),val(I2,2,T2),sim(T1,T2,S),S>90.
```

axioms

```
val(I1,2,N2):- eqv(I1,2,I2,2),loc(I2,2,N2).
i(I1,2,I2,2):- val(I1,2,V),val(I2,2,V), not empty(V).
eqo(X,Y):- eqo(Y,X). eqo(X,Y):- eqo(X,Z),eqo(Z,Y).
{eqo(X,Y)}:- activeo(X,Y).
```

cons

```
:- patient(I1,P1,_,_,_),patient(I2,P2,_,_,_),eqo(P1,P2), not i(I1,2,I2,2).
```

```
% facts
eqo(p3,p3).
eqv(t1,2,t1,2). sim(JohnSmith,JSmith,90).
…
loc(t1,2,JohnSmith). patient(t1,p1,JohnSmith,25,12345,brupen).

% rules
eqo(X,Y) :- patient(I1,X,A1,T1,_),patient(I2,Y,A2,T2,_),val(I1,2,N),val(I2,2,N).
activeo(X,Y) :- doctor(I1,X,_,D,T),doctor(I2,Y,_,D,T),
               app(I3,X1,P1),app(I4,Y1,P2), eqo(X1,X),eqo(Y1,Y), eqo(P1,P2).
eqv(I1,2,I2,2):- patient(I',P2,_,_,_),eqo(P1,P2),patient(I,P1,_,_,_),
               val(I1,2,T1),val(I2,2,T2),sim(T1,T2,S),S>90.

% axioms
val(I1,2,N2):- eqv(I1,2,I2,2),loc(I2,2,N2).
i(I1,2,I2,2):- val(I1,2,V),val(I2,2,V), not empty(V).
eqo(X,Y):- eqo(Y,X). eqo(X,Y):- eqo(X,Z),eqo(Z,Y).
{eqo(X,Y)}:- activeo(X,Y).

% cons
:- patient(I1,P1,_,_,_),patient(I2,P2,_,_,_),eqo(P1,P2), not i(I1,2,I2,2).
```

facts

```
eqo(p3,p3).
eqv(t1,2,t1,2). sim(JohnSmith,JSmith,90).
…
loc(t1,2,JohnSmith). patient(t1,p1,JohnSmith,25,12345,brupen).
```

rules

```
eqo(X,Y) :- patient(I1,X,A1,T1,_),patient(I2,Y,A2,T2,_),val(I1,2,N),val(I2,2,N).
activeo(X,Y) :- doctor(I1,X,_,D,T),doctor(I2,Y,_,D,T),
               app(I3,X1,P1),app(I4,Y1,P2), eqo(X1,X),eqo(Y1,Y), eqo(P1,P2).
eqv(I1,2,I2,2):- patient(I',P2,_,_,_),eqo(P1,P2),patient(I,P1,_,_,_),
               val(I1,2,T1),val(I2,2,T2),sim(T1,T2,S),S>90.
val(I1,2,N2):- eqv(I1,2,I2,2),loc(I2,2,N2).
```

axioms

```
i(I1,2,I2,2):- val(I1,2,V),val(I2,2,V), not empty(V).
eqo(X,Y):- eqo(Y,X). eqo(X,Y):- eqo(X,Z),eqo(Z,Y).
{eqo(X,Y)}:- activeo(X,Y).
```

cons

```
:- patient(I1,P1,_,_,_),patient(I2,P2,_,_,_),eqo(P1,P2), not i(I1,2,I2,2).
```

Optimality criteria

In a solution, wrt **set** or **cardinality**:

- The **more *Eq*** the **better**
- The **more rule applications (support)** the **better**

  *E.g. **{(a,b)}** could be derived from **hr1**, **hr2**. A sol have both **hr1**, **hr2** applied is better*

- The **less *absent* active pairs** (in active-facts but not in eq-facts) the **better**
- The **less soft rule violations** the **better**

  *E.g. **active(a,b), active(a,c)** are derived from **{sr1}**, **{sr1,sr2}**, a sol chooses **(a,b)** to be **false** is better*

Optimality criteria

In a solution, wrt **set** or **cardinality**:

- The **more *Eq*** the **better**
- The **more rule applications (support)** the **better**

  E.g. ***{(a,b)}*** *could be derived from* ***hr1***, ***hr2***. *A sol have both* ***hr1***, ***hr2*** *applied is better*

- The **less *absent* active pairs** (in active-facts but not in eq-facts) the **better**
- The **less soft rule violations** the **better**

  E.g. ***active(a,b), active(a,c)*** *are derived from* ***{sr1}***, ***{sr1,sr2}***, *a sol chooses* ***(a,b)*** *to be* ***false*** *is better*

First two are **the same under set optim**, so 7 different criteria overall

Computing optimal solutions

Formulation: **set (S)**, **cardinality/weight (C)** optimisation

- **Set**:
  - Set-inclusion preference (asprin)
  - Domain heuristic (heur)
- **Cardinality/weight** :
  - weight preference
  - weighted constraint (wc)
  - wc + heur
  - weighted constraint with 36 threads (wc-36)

Encoding example

The **more *Eq*** the **better**:

- **Set**: `#heuristic eq(X,Y). [1,true]` or asprin **superset** preference
- **Cardinality**: *weighted constraint* `#maximize{1@1,X,Y:eq(X,Y)}.`

The **more rule applications (support)** the **better**

- For each hard/soft rule assign a unique constant from {1,...,n} to the third argument e.g. `eqo(X,Y,1):- body(X,Y). activeo(X,Y,2):- body(X,Y).`
- Each eq-atom on axiom head is replaced by `eq(X,Y,0)`
- `#maximize{1@1,X,Y,I:eqo(X,Y,I),I!=0}.`

Experimental setup

Datasets:

- 6 **multi-table datasets** IMDB (movie), (**synthetic duplicates**) MUSIC, Pokemon
- Baselines ASPEn, 2 rule-based ER systems Magellan, JedAI

Name	#Rec	#Rel	#At	#Ref	#Dup	#Const
Imdb	30k	5	22	4	6k	64k
ImdbC	30k	5	22	4	6k	66k
Mu	41k	11	72	12	15k	156k
MuC	41k	11	72	12	15k	160k
MuCC	41k	11	72	12	15k	166k
Poke	240k	20	104	20	4k	349k

Table 4: Dataset Statistics. #-columns represent the number of records, relations, attributes, referential constraints and duplicates, respectively.

Performance

- Compared set inclusion (default) **MS-1** with **3 rule-based baselines**

Method	Data	F_1	(P	R)	t_o	t_g	t_s	#DC	Data	F_1	(P	R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80**	78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77**	71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40	95.67	18.78	N/A	N/A	0/5		97.49	99.40	95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39	**99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36	94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39	**99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39	**98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63	82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51**	38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46	59.30	105.06	N/A	N/A	0/23		32.75	73.95	21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45**	**95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86	**88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45**	**95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17	**88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62**	38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97	29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47	19.75	101.47	N/A	N/A	0/23		2.1	1.08	46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83	65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71	69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5	**88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73**	**74.03**	11,880	496.14	48.52	10/10

Performance

- Compared set inclusion (default) **MS-1** with **3 rule-based baselines**
- **Consistently better quality**

Method	Data	F_1	(P	R)	t_o	t_g	t_s	#DC	Data	F_1	(P	R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80**	78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**9.77**	71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40	95.67	18.78	N/A	N/A	0/5		97.49	9.40	95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39	**99.14**	610.49	**11.51**	**0.082**	1/5		96.99	9.36	94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39	**99.14**	86.24	13.02	0.096	3/5		**99.13**	9.39	**98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63	82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**7.51**	38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46	59.30	105.06	N/A	N/A	0/23		32.75	3.95	21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45**	**95.89**	666.01	**1.78**	**0.20**	21/23		90.31	1.86	**88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45**	**95.89**	665.65	1.82	0.21	23/23		**90.46**	2.17	**88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62**	38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97	29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47	19.75	101.47	N/A	N/A	0/23		2.1	1.08	46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83	65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71	69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5	**88.21**	993.31	97.51	23.02	23/23		**84.98**	**9.73**	**74.03**	11,880	496.14	48.52	10/10

Performance

- Compared set inclusion (default) **MS-1** with **3 rule-based baselines**
- **Consistently better quality**
- **Disadvantageous runtime** (*expensive preprocessing*)

Method	Data	F_1	(P	R)	t_o	t_g	t_s	#DC	Data	F_1	(P	R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80**	78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77**	71.12	**3.11**	N/A	N/A	0/5
JedAI	*Imdb*	97.49	99.40	95.67	18.78	N/A	N/A	0/5	*ImdbC*	97.49	99.40	95.67	18.67	N/A	N/A	0/5
ASPEN	*Imdb*	**99.27**	99.39	**99.14**	610.49	**11.51**	**0.082**	1/5	*ImdbC*	96.99	99.36	94.73	757.17	**11.75**	**0.088**	1/5
ASPEN+	*Imdb*	**99.27**	99.39	**99.14**	86.24	13.02	0.096	3/5	*ImdbC*	**99.13**	99.39	**98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63	82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51**	38.83	**66.87**	N/A	N/A	0/23
JedAI	*Mu*	70.67	87.46	59.30	105.06	N/A	N/A	0/23	*MuC*	32.75	73.95	21.02	101.02	N/A	N/A	0/23
ASPEN	*Mu*	**97.64**	**99.45**	**95.89**	666.01	**1.78**	**0.20**	21/23	*MuC*	90.31	91.86	**88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN+	*Mu*	**97.64**	**99.45**	**95.89**	665.65	1.82	0.21	23/23	*MuC*	**90.46**	92.17	**88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62**	38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97	29.74	260.96	N/A	N/A	0/10
JedAI	*MuCC*	31.04	72.47	19.75	101.47	N/A	N/A	0/23	*Poke*	2.1	1.08	46.56	**23.46**	N/A	N/A	0/10
ASPEN	*MuCC*	71.18	77.83	65.58	718.38	**21.01**	**10.47**	16/23	*Poke*	81.78	99.71	69.31	4.454	**311.37**	**0.91**	10/10
ASPEN+	*MuCC*	**88.85**	89.5	**88.21**	993.31	97.51	23.02	23/23	*Poke*	**84.98**	**99.73**	**74.03**	11,880	496.14	48.52	10/10

Performance

Method	Data	F_1	(P	/ R)	t_o	t_g	t_s	#DC	Data	F_1	(P	/ R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80**	78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77**	71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40	95.67	18.78	N/A	N/A	0/5		97.49	99.40	95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39	**99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36	94.73	757.17	**11.75**	**0.088**	1/5
ASPEN+		**99.27**	99.39	**99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39	**98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63	82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51**	38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46	59.30	105.06	N/A	N/A	0/23		32.75	73.95	21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45**	**95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86	**88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN+		**97.64**	**99.45**	**95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17	**88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62**	38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97	29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47	19.75	101.47	N/A	N/A	0/23		2.1	1.08	46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83	65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71	69.31	4,454	**311.37**	**0.91**	10/10
ASPEN+		**88.85**	89.5	**88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73**	**74.03**	11,880	496.14	48.52	10/10

Comparing optimality criteria

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxES/SS	90.50	92.20	**88.86**	**12.16**	50	1.86
	minAS	**91.88**	**95.11**	**88.86**	12.8	50	**1.7**
	minVS	91.7	94.73	88.85	12.44	50	1.78
MuCC	maxES/SS	88.85	89.5	**88.21**	23.02	50	1.67
	minAS	89.62	91.11	88.18	23.01	50	**1.54**
	minVS	**90.13**	**92.2**	88.15	**21.61**	50	2.48
Poke	maxES/SS	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	minAS	83.83	**99.73**	72.3	51.58	50	0.06
	minVS	84.62	**99.73**	73.48	56.87	50	**0.01**

set-optimisation

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxEC	90.51	92.20	**88.9**	35.09	50	**2.53**
	maxSC	90.52	92.21	**88.9**	**30.1**	16	12.09
	minAC	91.4	94.05	**88.9**	84.69	50	2.61
	minVC	**92.01**	**95.35**	88.89	48.97	50	5.66
MuCC	maxEC	88.83	89.45	**88.21**	66.71	1	N/A
	maxSC	88.85	89.50	**88.21**	**52.8**	2	629.56
	minAC	89.51	90.93	88.14	92.97	50	**2.12**
	minVC	**89.74**	**91.36**	88.18	67.82	50	8.42
Poke	maxEC	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	maxSC	**84.98**	**99.73**	**74.03**	49.1	1	N/A
	minAC	84.80	**99.73**	73.75	49.23	50	**0.037**
	minVC	84.91	**99.73**	73.91	49.11	2	1.15

cardinality-optimisation

Comparing optimality criteria

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxES/SS	90.50	92.20	**88.86**	**12.16**	50	1.86
	minAS	**91.88**	**95.11**	**88.86**	12.8	50	**1.7**
	minVS	91.7	94.73	88.85	12.44	50	1.78
MuCC	maxES/SS	88.85	89.5	**88.21**	23.02	50	1.67
	minAS	89.62	91.11	88.18	23.01	50	**1.54**
	minVS	**90.13**	**92.2**	88.15	**21.61**	50	2.48
Poke	maxES/SS	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	minAS	83.83	**99.73**	72.3	51.58	50	0.06
	minVS	84.62	**99.73**	73.48	56.87	50	**0.01**

set-optimisation

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxEC	90.51	92.20	**88.9**	35.09	50	**2.53**
	maxSC	90.52	92.21	**88.9**	**30.1**	16	12.09
	minAC	91.4	94.05	**88.9**	84.69	50	2.61
	minVC	**92.01**	**95.35**	88.89	48.97	50	5.66
MuCC	maxEC	88.83	89.45	**88.21**	66.71	1	N/A
	maxSC	88.85	89.50	**88.21**	**52.8**	2	629.56
	minAC	89.51	90.93	88.14	92.97	50	**2.12**
	minVC	**89.74**	**91.36**	88.18	67.82	50	8.42
Poke	maxEC	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	maxSC	**84.98**	**99.73**	**74.03**	49.1	1	N/A
	minAC	84.80	**99.73**	73.75	49.23	50	**0.037**
	minVC	84.91	**99.73**	73.91	49.11	2	1.15

cardinality-optimisation

Comparing optimality criteria

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxES/SS	90.50	92.20	**88.86**	**12.16**	50	1.86
	minAS	**91.88**	**95.11**	**88.86**	12.8	50	**1.7**
	minVS	91.7	94.73	88.85	12.44	50	1.78
MuCC	maxES/SS	88.85	89.5	**88.21**	23.02	50	1.67
	minAS	89.62	91.11	88.18	23.01	50	**1.54**
	minVS	**90.13**	**92.2**	88.15	**21.61**	50	2.48
Poke	maxES/SS	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	minAS	83.83	**99.73**	72.3	51.58	50	0.06
	minVS	84.62	**99.73**	73.48	56.87	50	**0.01**

Solving with **single** threads

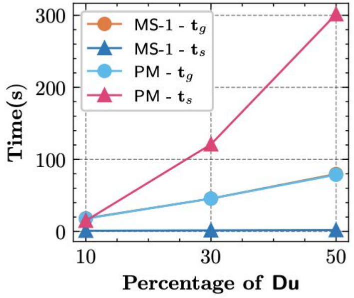
Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxEC	90.51	92.20	**88.9**	35.09	50	**2.53**
	maxSC	90.52	92.21	**88.9**	**30.1**	16	12.09
	minAC	91.4	94.05	**88.9**	84.69	50	2.61
	minVC	**92.01**	**95.35**	88.89	48.97	50	5.66
MuCC	maxEC	88.83	89.45	**88.21**	66.71	1	N/A
	maxSC	88.85	89.50	**88.21**	**52.8**	2	629.56
	minAC	89.51	90.93	88.14	92.97	50	**2.12**
	minVC	**89.74**	**91.36**	88.18	67.82	50	8.42
Poke	maxEC	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	maxSC	**84.98**	**99.73**	**74.03**	49.1	1	N/A
	minAC	84.80	**99.73**	73.75	49.23	50	**0.037**
	minVC	84.91	**99.73**	73.91	49.11	2	1.15

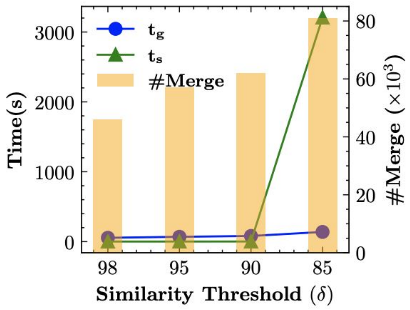
Solving with **36** threads!

Factors impacting scalability

- **While keeping other 2 factors the same, varying**:
    - **Datasize |D|** -> the **larger** the **slower**
    - **% of duplicates (Du%)** -> the **higher** the **slower**
    - **Similarity thresholds** -> the **lower** the **slower**
- **Grounding time** is more sensitive to **datasize**, **similarity**
- **Solving time** is more sensitive to **% of duplicates** and **similarity**

$\|D\|$	Met.	t_g	t_s	Met.	t_g	t_s
×1	MS-1	17.82	1.02	PM	17.55	14.87
×2		92.4	2.21		92.75	84.6
×3		388.2	3.4		418.17	228.87
×4		719.2	4.8		763.61	416.48
×5		1083.6	5.7		1106.7	735.43





Conclusion

- Introduced **ASPEn+**, an **ASP-based system for collective ER**, first of its kind that supports both **global** and **local** merges
- Defined **seven optimality criteria** for **preferred** ER solutions, and provided **complexity analyses**.
- Conducted extensive experiments, achieving **superior accuracy** on complex, **real-world** datasets, demonstrating the practical benefits of **local semantics** and **flexible optimisation**

Future work

- Computing **brave**/**cautious** consequences over space of optimal solutions
    - Meta programming *[Gebser et al, TPLP'2011]*
    - WASP *[Alviano et al, AIJ'2023]*
- Learning rules (**ILP style**, ASP-based meta program)
    - Learning from Failure (Popper ILP system) *[Cropper et al, MLJ' 2021]*