Big Data Cleaning

Paolo Papotti
EURECOM, France

3rd International KEYSTONE Conference 2017

every MINUTE of the DAY

Skype USERS MAKE
110,040 CALLS

UBER
PASSENGERS TAKE **694** RIDES

Facebook
USERS LIKE
4,166,667 POSTS

TWITTER USERS SEND
347,222 TWEETS

YOUTUBE
USERS UPLOAD
300 HOURS
OF NEW VIDEO

Buzzfeed
USERS VIEW
34,150 VIDEOS

SNAPCHAT USERS SHARE
284,722 SNAPS

Instagram
USERS LIKE
1,736,111 PHOTOS

Tinder
USERS SWIPE
590,278 TIMES

PINTEREST
USERS PIN
9,722 IMAGES

VINE USERS PLAY
1,041,666 VIDEOS

AMAZON
RECEIVES
4,310
UNIQUE VISITORS

reddit
USERS CAST
18,327 VOTES

Netflix
SUBSCRIBERS STREAM
77,160
HOURS OF VIDEO

APPLE USERS DOWNLOAD
51,000 APPS

Home | Applications | Analyze

Cyber Attacks by Actors Related to Iran

ScarCruft – Threat Actor

1 000+ References to This Entity
First Seen Jun 7, 2016
Last Seen Jun 23, 2016
Curated Entity
Category Nation State Sponsored (APT)

Print
Request Data Review
Add to List
EXPORT ENTITIES

Show all events involving ScarCruft in Table

Total Reference Count

2 223 Total References
2 223 In the Last 60 Days
1 723 In the Last 7 Days
3 References Today

References Breakdown

947 In Social Media
2 144 From Information Security Sources
1 582 Including Malicious Language

Show recent events in Table

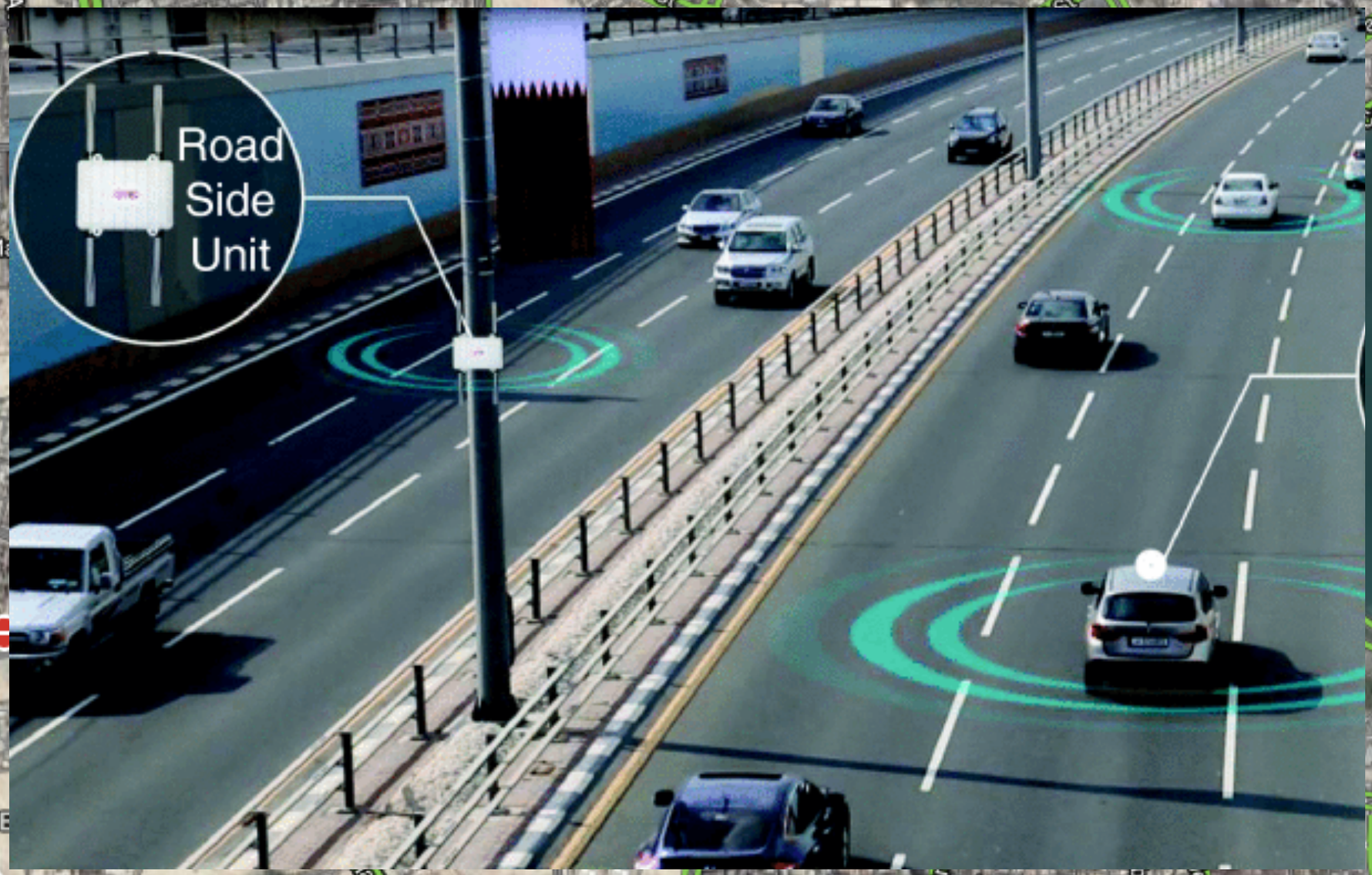
Attacker Directly Mentioned in Cyber Attacks

945 Total References
945 References in the Last 60 Days (Including Future Events)
726 In the Last 7 Days
0 References Today

Show recent cyber events in Table

United Sta
Israel
Saudi Arab
Iran
Turkey
Qatar
United Kin
Nepal
Other
Jan 20
Jan 2013
Jan 2014
Jun 11 2010
5 years
Jun 11 2015

up to 26% errors
[Abedjan et al, 2015]



5% measurement errors
7% duplicate devices
sensors with up to 30% errors

Is quality of data important?

- Many decisions are taken after manually scrutinizing the data
  - Military attack
- But more and more are taken by algorithms
  - Stocks trading
  - Credit report/Risk assessment
  - Self driving cars

But it is expensive!

The New York Times

TECHNOLOGY

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

By STEVE LOHR AUG. 17, 2014

Yet far too much handcrafted work — what data scientists call "data wrangling," "data munging" and "data janitor work" — is still required. Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

Data quality facts

Recorded Future
CREATING AN INSIGHTFUL WORLD

"**engineers** dedicated to data integration and cleaning" [CIO]

"**50 people** curating products' data"
[Chief scientist]

@WalmartLabs

verizon

"Typical duration of an integration project is in terms of **years**"
[Former Chief Scientist]

Google Cloud Platform [https://cloud.google.com/dataprep]

Search

Why Google | Products | Solutions | Launcher | Pricing | Customers | Documentation | Support | Partners

GOOGLE CLOUD BIG DATA AND MACHINE LEARNING BLOG

Innovation in data processing and machine learning technology

Google Cloud Platform adds new tools for easy data preparation and integration

Thursday, March 9, 2017

Big Data

Product dee
technical co

FORRESTER®

FOR CUSTOMER INSIGHTS PROFESSIONALS

The Forrester Wave™: Data Preparation Tools, Q1 2017

The Seven Providers That Matter Most And How They Stack Up

by Cinny Little
March 13, 2017

Google Cloud Platform [https://cloud.google.com/dataprep]

Search

Why Google | Products | Solutions | Launcher | Pricing | Customers | Documentation | Support | Partners

GOOGLE CLOUD BIG DATA AND MACHINE LEARNING BLOG

Innovation in data processing and machine learning technology

Google Cloud Platform adds new tools for easy data preparation and integration

Thursday, March 9, 2017

Big Data

Product dee

technical co

FORRESTER®

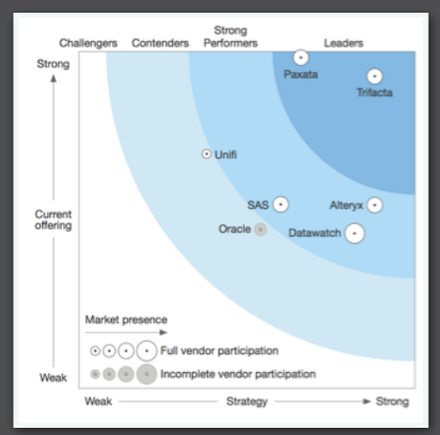
FOR CUSTOMER INSIGHTS PROFESSIONALS

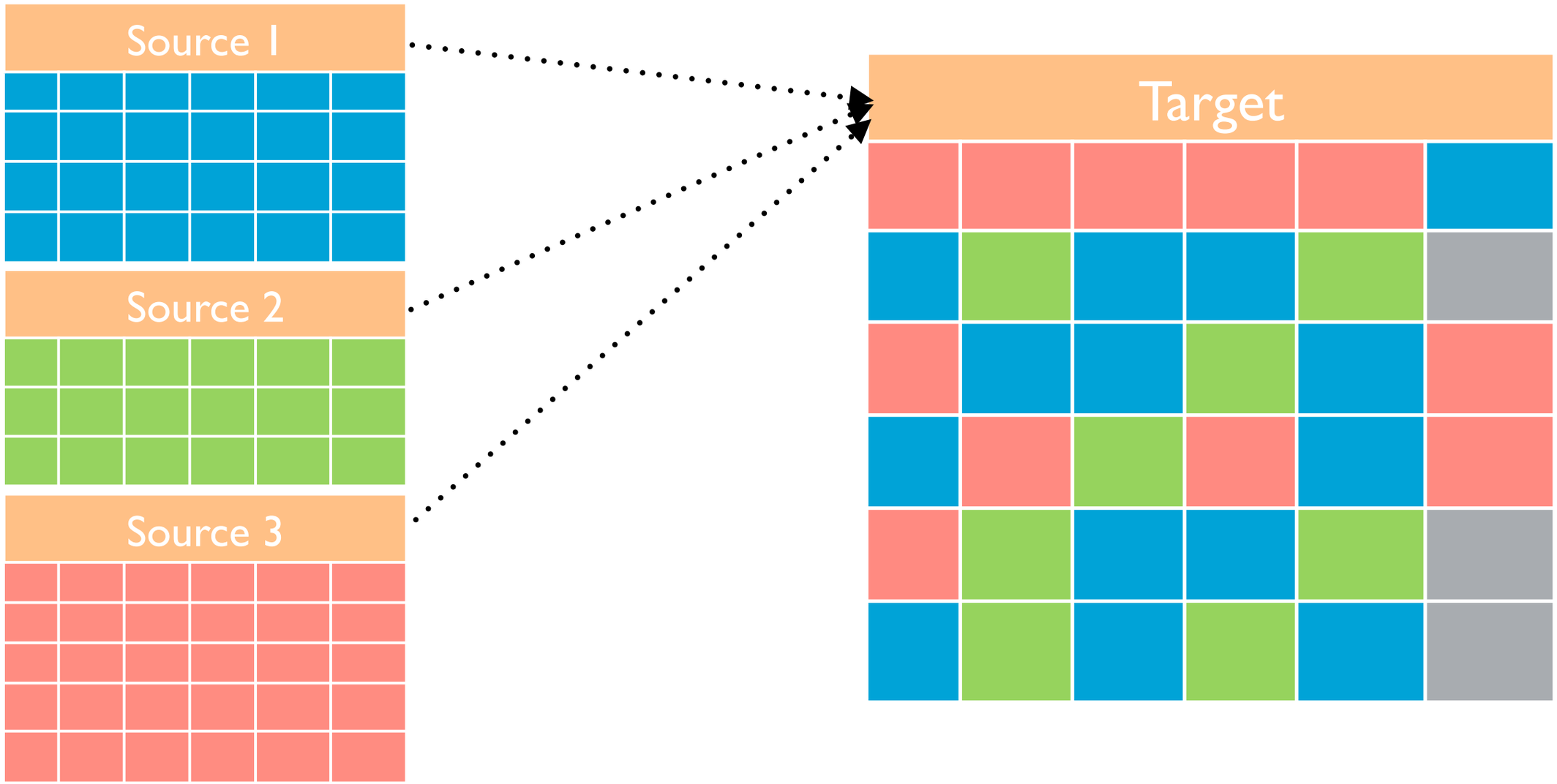
The Forrester Wave™: Data Preparation Tools, Q1 2017

The Seven Providers That Matter Most And How They Stack Up

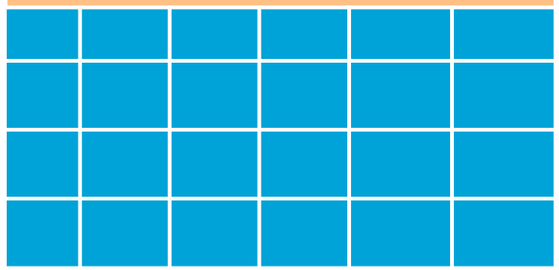
by Cinny Little

March 13, 2017

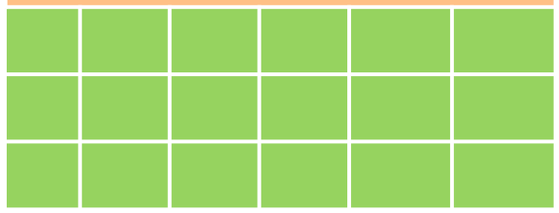



Source 1
Source 2
Source 3
Target

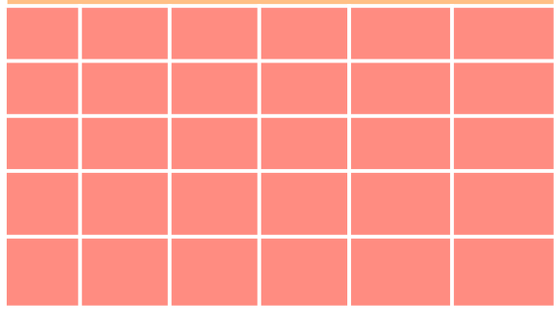
Source 1



Source 2



Source 3



Target

```
BEGIN TRANSACTION;
SET CONSTRAINTS ALL DEFERRED;delete from target.PersonSet;delete from target.CarSet;delete from target.MakeSet;delete from target.CitySet;
------------------------------  TGDS  ----------------------------------
create table work.TARGET_VALUES_TGD_v8_v3 AS
    select distinct          null as v3id,  rel_v8.cityName as v3name,  rel_v8.region as v3region
    from source.CityRegionSet AS rel_v8;
create table work.TARGET_VALUES_TGD_v5_v0v1 AS
    select distinct          null as v0id,          rel_v5.personName as v0name,          null as v0age,
        'SK{T=ll'[0.0:'llrel_v5.personNamell']'ll'-ll'[1.1:'llrel_v5.carModelll']'ll'J=ll'[ll'[0.0:'llrel_v5.personNamell']'ll'.0.2'll'-ll'[1.1:'llrel_v5.carModelll']'ll'.1.3'll'V=ll'[ll'0.2'll'-ll'1.3'll'}'
as v0carId,          null as v0cityId,
        'SK{T=ll'[0.0:'llrel_v5.personNamell']'ll'-ll'[1.1:'llrel_v5.carModelll']'ll'J=ll'[ll'[0.0:'llrel_v5.personNamell']'ll'.0.2'll'-ll'[1.1:'llrel_v5.carModelll']'ll'.1.3'll'V=ll'[ll'0.2'll'-ll'1.3'll'}'
as v1id,          rel_v5.carModel as v1model,          null as v1plate,          null as v1makeId
    from source.PersonCarSet2 AS rel_v5;
create table work.TARGET_VALUES_TGD_v6_v0v3 AS
    select distinct          null as v0id,          rel_v6.personName as v0name,          null as v0age,          null as v0carId,
        'SK{T=ll'[0.0:'llrel_v6.personNamell']'ll'-ll'[2.4:'llrel_v6.cityNamell']'ll'J=ll'[ll'[0.0:'llrel_v6.personNamell']'ll'.0.5'll'-ll'[2.4:'llrel_v6.cityNamell']'ll'.2.6'll'V=ll'[ll'0.5'll'-ll'2.6'll'}'
as v0cityId,
        'SK{T=ll'[0.0:'llrel_v6.personNamell']'ll'-ll'[2.4:'llrel_v6.cityNamell']'ll'J=ll'[ll'[0.0:'llrel_v6.personNamell']'ll'.0.5'll'-ll'[2.4:'llrel_v6.cityNamell']'ll'.2.6'll'V=ll'[ll'0.5'll'-ll'2.6'll'}'
as v3id,          rel_v6.cityName as v3name,          null as v3region
    from source.PersonCitySet AS rel_v6;
create table work.TARGET_VALUES_TGD_v7_v1v2 AS
    select distinct          null as v1id,          rel_v7.carModel as v1model,          null as v1plate,
        'SK{T=ll'[1.1:'llrel_v7.carModelll']'ll'-ll'[3.7:'llrel_v7.makeNamell']'ll'J=ll'[ll'[1.1:'llrel_v7.carModelll']'ll'.1.8'll'-ll'[3.7:'llrel_v7.makeNamell']'ll'.3.9'll'V=ll'[ll'1.8'll'-ll'3.9'll'}' as
v1makeId,
        'SK{T=ll'[1.1:'llrel_v7.carModelll']'ll'-ll'[3.7:'llrel_v7.makeNamell']'ll'J=ll'[ll'[1.1:'llrel_v7.carModelll']'ll'.1.8'll'-ll'[3.7:'llrel_v7.makeNamell']'ll'.3.9'll'V=ll'[ll'1.8'll'-ll'3.9'll'}' as
v2id,
        rel_v7.makeName as v2name
    from source.CarMakeSet AS rel_v7;
create table work.TARGET_VALUES_TGD_v4_v0v1 AS
    select distinct          null as v0id,          rel_v4.personName as v0name,          rel_v4.age as v0age,
        'SK{T=ll'[0.0:'llrel_v4.personNamell'-ll'0.10:'llrel_v4.agell']'ll'-ll'[1.11:'llrel_v4.carPlatell']'ll'J=ll'[ll'[0.0:'llrel_v4.personNamell'-ll'0.10:'llrel_v4.agell']'ll'.0.2'll'-ll'[1.11:'ll
rel_v4.carPlatell']'ll'.1.3'll'V=ll'[ll'0.2'll'-ll'1.3'll'}' as v0carId,          null as v0cityId,
        'SK{T=ll'[0.0:'llrel_v4.personNamell'-ll'0.10:'llrel_v4.agell']'ll'-ll'[1.11:'llrel_v4.carPlatell']'ll'J=ll'[ll'[0.0:'llrel_v4.personNamell'-ll'0.10:'llrel_v4.agell']'ll'.0.2'll'-ll'[1.11:'ll
rel_v4.carPlatell']'ll'.1.3'll'V=ll'[ll'0.2'll'-ll'1.3'll'}' as v1id,
        null as v1model,          rel_v4.carPlate as v1plate,          null as v1makeId
    from source.PersonCarSet1 AS rel_v4;
----------------------  RESULT OF EXCHANGE --------------------------
insert into target.PersonSet
    select          cast(work.TARGET_VALUES_TGD_v4_v0v1.v0id as text) as v0id,          cast(work.TARGET_VALUES_TGD_v4_v0v1.v0name as text) as v0name,
        cast(work.TARGET_VALUES_TGD_v4_v0v1.v0age as text) as v0age,          cast(work.TARGET_VALUES_TGD_v4_v0v1.v0carId as text) as v0carId,
cast(work.TARGET_VALUES_TGD_v4_v0v1.v0cityId as text) as v0cityId
    from          work.TARGET_VALUES_TGD_v4_v0v1
    UNION
    select          cast(work.TARGET_VALUES_TGD_v6_v0v3.v0id as text) as v0id,          cast(work.TARGET_VALUES_TGD_v6_v0v3.v0name as text) as v0name,
        cast(work.TARGET_VALUES_TGD_v6_v0v3.v0age as text) as v0age,          cast(work.TARGET_VALUES_TGD_v6_v0v3.v0carId as text) as v0carId,
        cast(work.TARGET_VALUES_TGD_v6_v0v3.v0cityId as text) as v0cityId
    from          work.TARGET_VALUES_TGD_v6_v0v3
    UNION
    select          cast(work.TARGET_VALUES_TGD_v5_v0v1.v0id as text) as v0id,          cast(work.TARGET_VALUES_TGD_v5_v0v1.v0name as text) as v0name,
        cast(work.TARGET_VALUES_TGD_v5_v0v1.v0age as text) as v0age,          cast(work.TARGET_VALUES_TGD_v5_v0v1.v0carId as text) as v0carId,
cast(work.TARGET_VALUES_TGD_v5_v0v1.v0cityId as text) as v0cityId
    from          work.TARGET_VALUES_TGD_v5_v0v1;
insert into target.CarSet
    select          cast(work.TARGET_VALUES_TGD_v4_v0v1.v1id as text) as v1id,          cast(work.TARGET_VALUES_TGD_v4_v0v1.v1model as text) as v1model,
cast(work.TARGET_VALUES_TGD_v4_v0v1.v1plate as text) as v1plate,
        cast(work.TARGET_VALUES_TGD_v4_v0v1.v1makeId as text) as v1makeId
    from          work.TARGET_VALUES_TGD_v4_v0v1
    UNION
    select          cast(work.TARGET_VALUES_TGD_v7_v1v2.v1id as text) as v1id,          cast(work.TARGET_VALUES_TGD_v7_v1v2.v1model as text) as v1model,
cast(work.TARGET_VALUES_TGD_v7_v1v2.v1plate as text) as v1plate,
        cast(work.TARGET_VALUES_TGD_v7_v1v2.v1makeId as text) as v1makeId
    from          work.TARGET_VALUES_TGD_v7_v1v2
    UNION
    select          cast(work.TARGET_VALUES_TGD_v5_v0v1.v1id as text) as v1id,          cast(work.TARGET_VALUES_TGD_v5_v0v1.v1model as text) as v1model,
cast(work.TARGET_VALUES_TGD_v5_v0v1.v1plate as text) as v1plate,          cast(work.TARGET_VALUES_TGD_v5_v0v1.v1makeId as text) as v1makeId
    from          work.TARGET_VALUES_TGD_v5_v0v1;
insert into target.MakeSet
    select          cast(work.TARGET_VALUES_TGD_v7_v1v2.v2id as text) as v2id,          cast(work.TARGET_VALUES_TGD_v7_v1v2.v2name as text) as v2name
    from          work.TARGET_VALUES_TGD_v7_v1v2;
insert into target.CitySet
    select          cast(work.TARGET_VALUES_TGD_v6_v0v3.v3id as text) as v3id,          cast(work.TARGET_VALUES_TGD_v6_v0v3.v3name as text) as v3name,
cast(work.TARGET_VALUES_TGD_v6_v0v3.v3region as text) as v3region
    from          work.TARGET_VALUES_TGD_v6_v0v3
    UNION
```

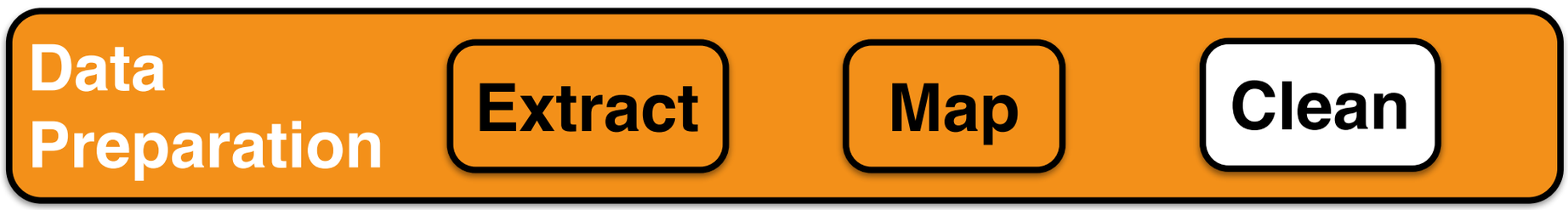
Declarative Approach

1. Formalization

clear notion of **desired** solution

2. Scalable algorithms

handle **large** datasets



Data Cleaning

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

Up to 25% of **business**, **health**, and **scientific** data is dirty: **errors**, **missing values**, **duplicates**

[https://www.gartner.com/doc/3169421/magic-quadrant-data-quality-tools]

Data Cleaning

ID	FN	LN	ROLE	ZIP	ST	SAL
~~105~~	~~Anne~~	~~Nash~~	~~E~~	~~85281~~	~~NY~~	~~110~~
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

- One declarative approach based on rules
  - Functional Dependency: zip code identifies state
- A **repair** is an updated, consistent instance

Data Cleaning

- **Computing an optimal repair is a NP problem**

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	**AZ**	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

- One declarative approach based on rules
  - Functional Dependency: zip code identifies state
- A **repair** is an updated, consistent instance
- An **optimal repair** is minimal in terms of number of changes between the original dataset and the repair

Data Cleaning

- Computing an **optimal repair** is a NP problem

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

- **Multiple possible ways** to repair a violation
- **Domino effect**: new violations could be generated by resolving a violation [Xu et al, 2013a]
- Approximate solution with heuristics

Rule Based Data Cleaning

- Functional dependencies [Bohannon et al, 2005], Conditional Function Dependencies [Cong et al, 2007], Conditional Inclusion Dependencies [Bravo et al, 2007], Matching Dependencies [Bertossi et al, 2011], Editing Rules [Fan et al, 2010], Fixing Rules [Tang, 2014]

- Each fragment covers a new aspect:
  axioms, complexity study, heuristic repair algorithm
- Sequence of repair algorithms: poor repair
  - **0.3 F-measure over real data**

- Piecemeal approach misses evidence!

Denial Constraints (DCs)

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

Denial Constraints (DCs)

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“}M\text{”} \wedge t_\beta.ROLE = \text{“}E\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Denial Constraints (DCs)

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“}M\text{”} \wedge t_\beta.ROLE = \text{“}E\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Denial Constraints (DCs)

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“}M\text{”} \wedge t_\beta.ROLE = \text{“}E\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Denial Constraints (DCs)

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“}M\text{”} \wedge t_\beta.ROLE = \text{“}E\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Denial Constraints (DCs)

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

repair condition

$$t_\alpha.ST = t_\beta.ST$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“}M\text{”}$$
$$\wedge t_\beta.ROLE = \text{“}E\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Denial Constraints (DCs)

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

repair condition

$$t_\alpha.ST = t_\beta.ST$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“}M\text{”} \wedge t_\beta.ROLE = \text{“}E\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Denial Constraints (DCs)

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	E	85281	NY	110
211	Mark	White	M	15544	NY	80
386	Mark	Lee	E	85281	AZ	75
215	Anna	Smith Nash	E	85283		

repair condition

$t_\alpha.ST = t_\beta.ST$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“}M\text{”} \wedge t_\beta.ROLE = \text{“}E\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

$t_\alpha.ST \neq t_\beta.ST$

Two Steps for Cleaning

- Detect:
  identify constraint **violations**
- Repair:
  identify **errors** and suggest **repairs**
  - idea: exploit *interactions* among violations for better repairs

Conflict Hypergraph

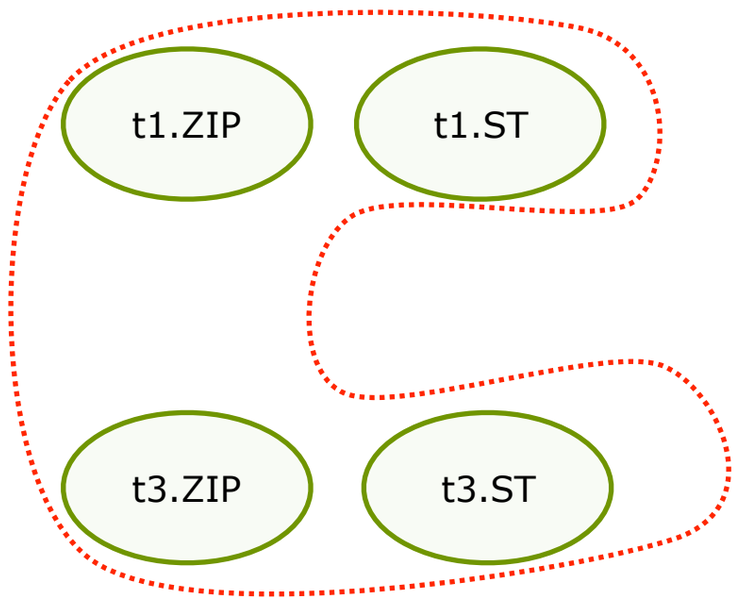
	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85281	NY	110
t_2	211	Mark	White	M	15544	NY	80
t_3	386	Mark	Lee	E	85281	AZ	75
t_4	215	Anna	Smith Nash	E	85283		

[Xu et al, 2013a]

Conflict Hypergraph

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85281	NY	110
t_2	211	Mark	White	M	15544	NY	80
t_3	386	Mark	Lee	E	85281	AZ	75
t_4	215	Anna	Smith Nash	E	85283		

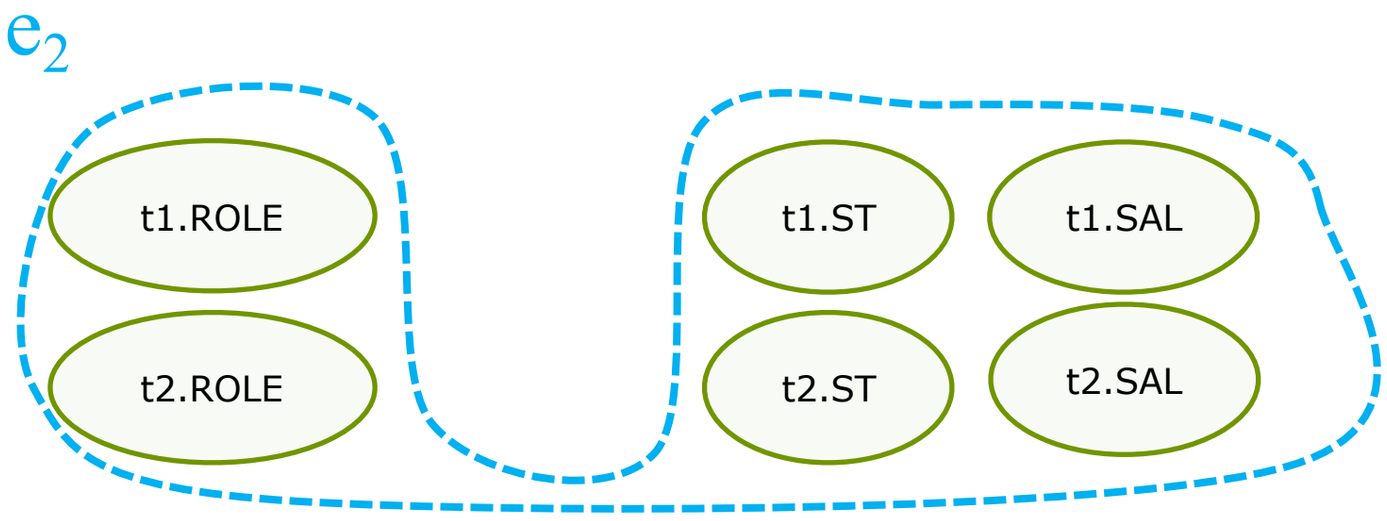
e_1



[Xu et al, 2013a]

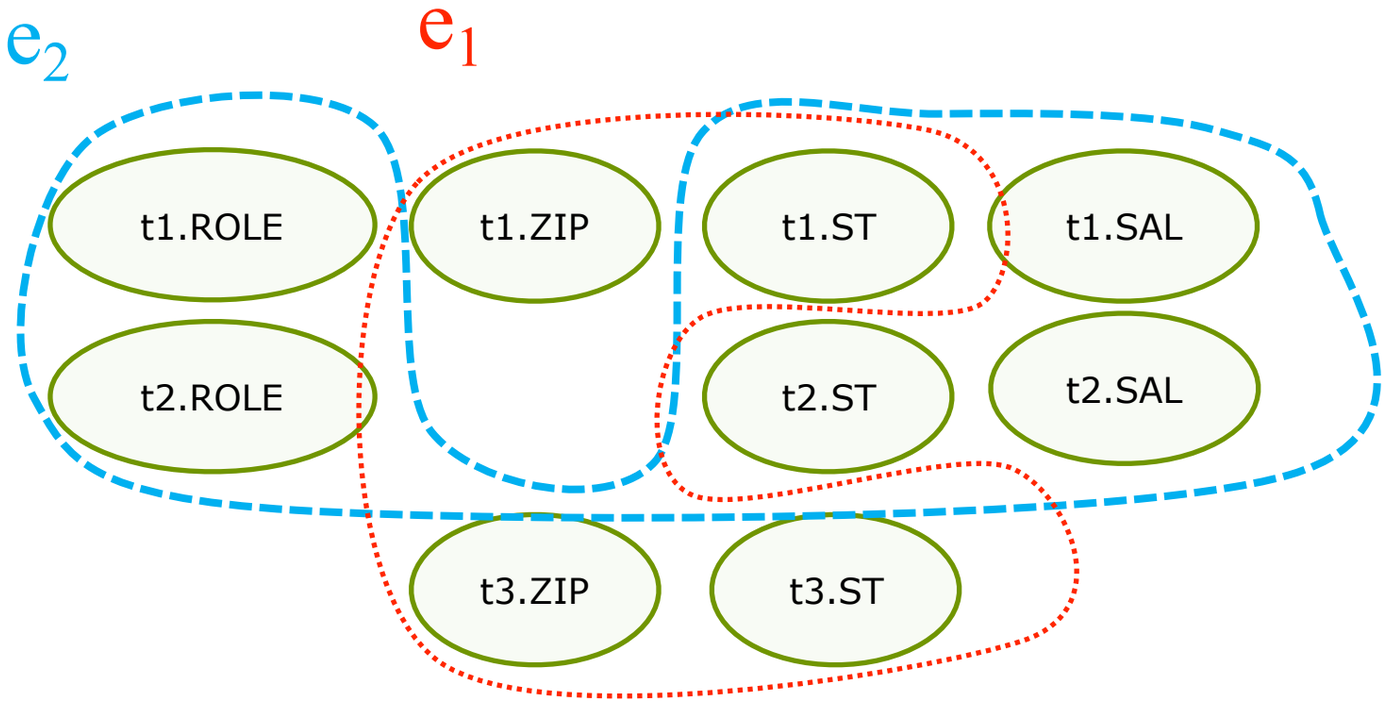
Conflict Hypergraph

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85281	NY	110
t_2	211	Mark	White	M	15544	NY	80
t_3	386	Mark	Lee	E	85281	AZ	75
t_4	215	Anna	Smith Nash	E	85283		



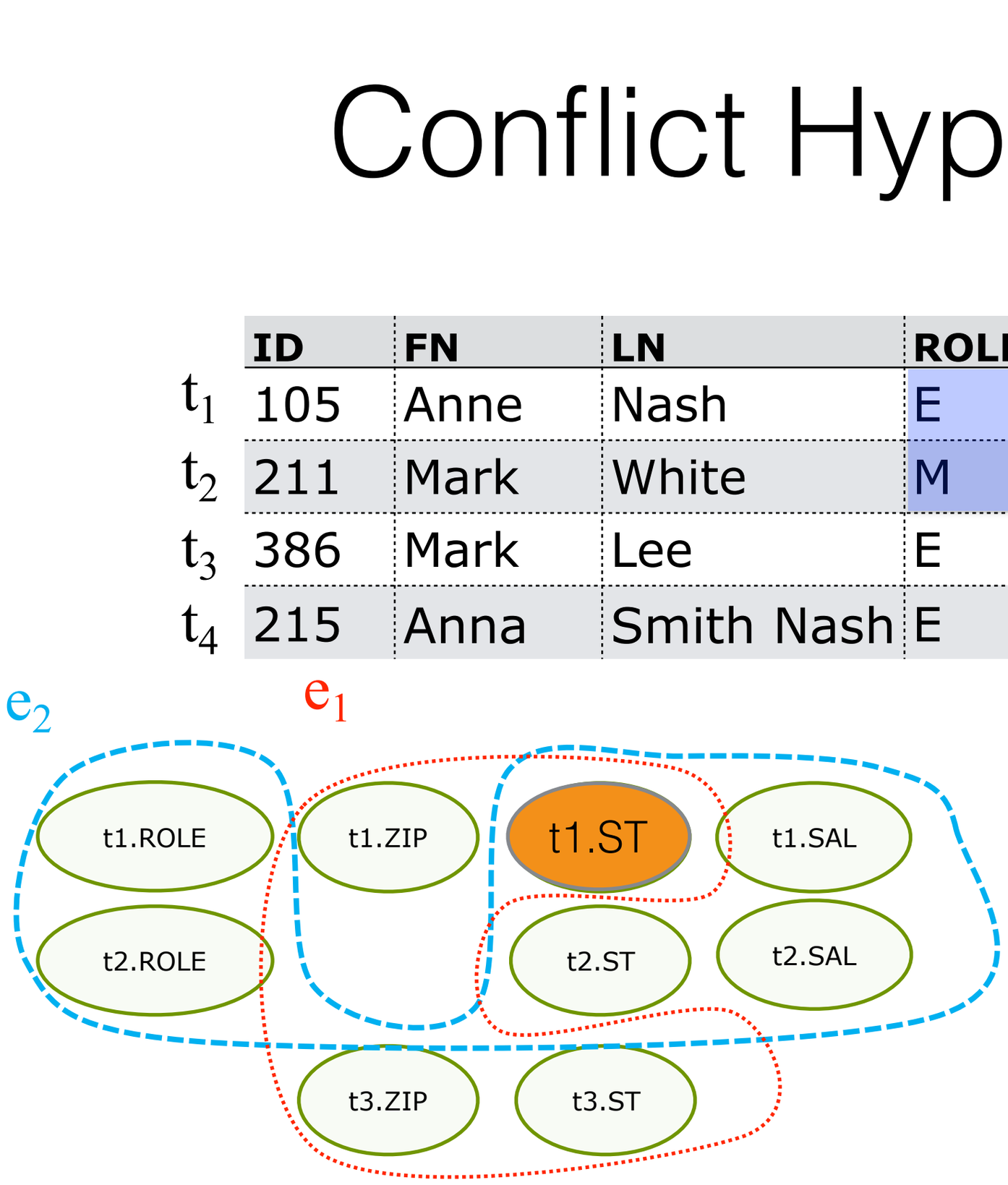
Conflict Hypergraph

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85281	NY	110
t_2	211	Mark	White	M	15544	NY	80
t_3	386	Mark	Lee	E	85281	AZ	75
t_4	215	Anna	Smith Nash	E	85283		



Conflict Hypergraph

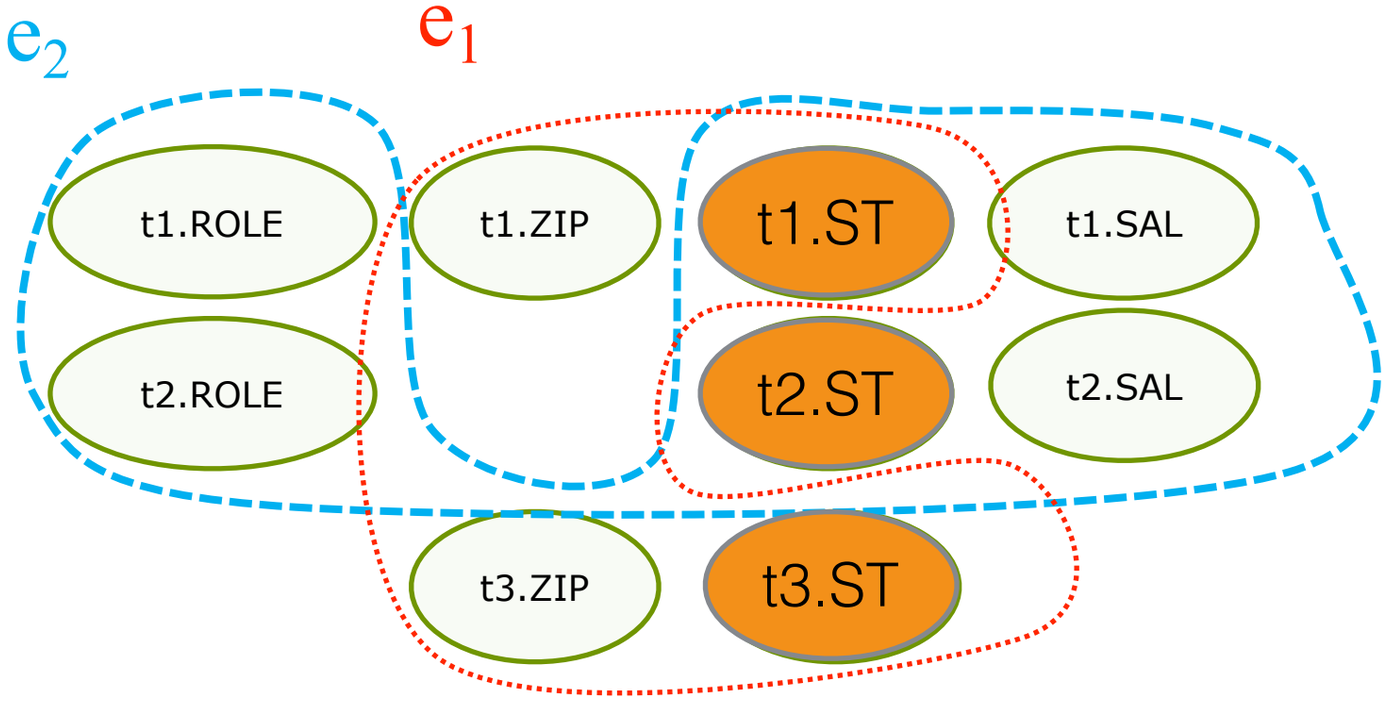
	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85281	NY	110
t_2	211	Mark	White	M	15544	NY	80
t_3	386	Mark	Lee	E	85281	AZ	75
t_4	215	Anna	Smith Nash	E	85283		



- MVC: t1.ST

Conflict Hypergraph

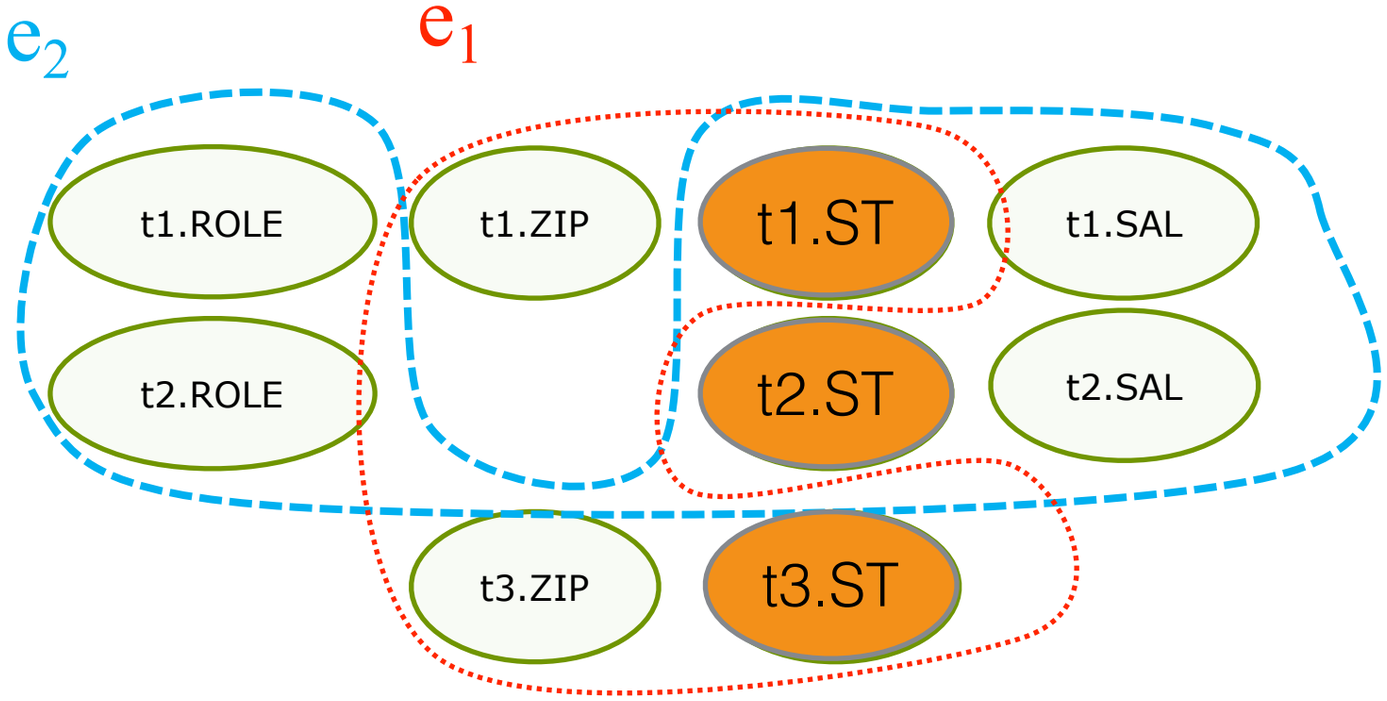
	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85281	NY	110
t_2	211	Mark	White	M	15544	NY	80
t_3	386	Mark	Lee	E	85281	AZ	75
t_4	215	Anna	Smith Nash	E	85283		



- MVC: t1.ST
- system:
  t1.ST != t2.ST
  t1.ST = t3.ST

Conflict Hypergraph

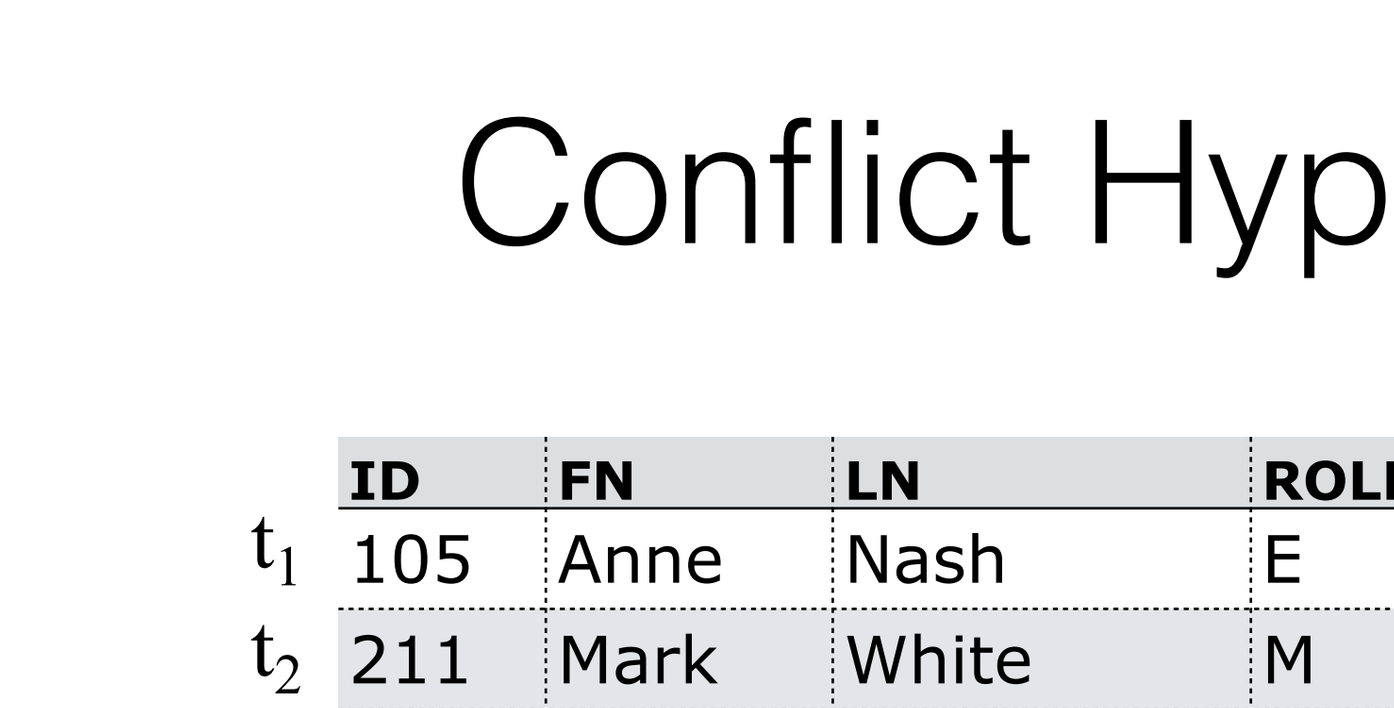
	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85281	**AZ**	110
t_2	211	Mark	White	M	15544	NY	80
t_3	386	Mark	Lee	E	85281	AZ	75
t_4	215	Anna	Smith Nash	E	85283		



- MVC: t1.ST
- system:
  t1.ST != t2.ST
  t1.ST = t3.ST
- update and iterate

Conflict Hypergraph

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85281	**AZ**	110
t_2	211	Mark	White	M	15544	NY	80
t_3	386	Mark	Lee	E	85281	AZ	75
t_4	215	Anna	Smith Nash	E	85283		



- MVC: t1.ST
- system:
  t1.ST != t2.ST
  t1.ST = t3.ST
- update and iterate

- Th: constant factor approx. algorithm

Experimental Results: DCs

- Nine datasets, 4000 manually annotated tuples

		P	R	F
Company Employees #	24	0.74	0.17	**0.27**
Company Meet.	336	0.94	0.5	**0.65**
Credit Rating	48	0.6	0.75	**0.67**
Employment Change	24	1.0	0.88	**0.94**
Natural Disaster	24	0.8	0.5	0.62
Person Travel	48	0.61	0.82	0.7
Political Endorsement	48	1.0	0.59	**0.74**
Product Recall	177	0.9	0.9	**0.9**
Voting Result	24	1.0	0.6	**0.75**
		0.84	0.54	

[Abedjan et al, 2015]

Cleaning with Denial Constraints

- Language: axioms, implication testing
- Semantics: partial order over groups of values
- Algorithms: constant factor approximation
- System: scalable, disk-based cleaning tools

Users define the rules: model for the background knowledge to be enforced on the data

Supporting Rules Discovery

- Large literature on Functional Dependencies [Kivinen and Mannila, 1995]
- More recent efforts on data quality rules
  - Conditional Functional Dependencies [Chiang and Miller, 2008]
  - Matching Dependencies [Song and Chen, 2009]
  - Denial Constraints [Xu et al, 2013b]

Discovering DCs

DATASET
Tax

Browse...

Approximate Threshold: 0.01

Constant Frequency: 0

Go

Formula | Linguistics

Coverage : 0.40

Succinctness: 0.60

Filtering:

FDs

not(t1.areacode=t2.areacode & t1.phone=t2.phone)	Yes	No
not(t1.city!=t2.city & t1.zip=t2.zip)	Yes	No
not(t1.state=t2.state & t1.haschild=t2.haschild & t1.childexemp!=t2.childexemp)	Yes	No
not(t1.state=t2.state & t1.maritalstatus=t2.maritalstatus & t1.singleexemp!=t2.singleexemp)	Yes	No
not(t1.state=t2.state & t1.salary=t2.salary & t1.rate!=t2.rate)	Yes	No
not(t1.state=t2.state & t1.salary>t2.salary & t1.rate<t2.rate)	Yes	No
not(t1.phone=t2.phone)	Yes	No
not(t1.fname=t2.fname)	Yes	No

There cannot exist two tuples tα, tβ in the dataset, such that they have different city ,and they have same zip

Data | Example

Negative Example:

tid	fname	lname	areacode	phone	city	state	zip	maritalstatus	haschild	salary	rate	sing
1	Mark	Ballin	304	2327667	Anthony	AR	25813	S	Y	5000	3	200
8	Marcelino	Nuth	304	5404707	Kyle	WV	25813	M	N	10000	4	0

Positive Examples:

tid	fname	lname	areacode	phone	city	state	zip	maritalstatus	haschild	salary	rate	sing
1	Mark	Ballin	304	2327667	Anthony	WV	25813	S	Y	5000	3	200
8	Marcelino	Nuth	304	5404707	Kyle	WV	25813	M	N	10000	4	0

tid	fname	lname	areacode	phone	city	state	zip	maritalstatus	haschild	salary	rate	sing
1	Mark	Ballin	304	2327667	Anthony	AR	25813	S	Y	5000	3	200
8	Marcelino	Nuth	304	5404707	Kyle	AR	25813	M	N	10000	4	0

tid	fname	lname	areacode	phone	city	state	zip	maritalstatus	haschild	salary	rate	sing
1	Mark	Ballin	304	2327667	Anthony	AR	10000	S	Y	5000	3	200
8	Marcelino	Nuth	304	5404707	Kyle	WV	25813	M	N	10000	4	0

tid	fname	lname	areacode	phone	city	state	zip	maritalstatus	haschild	salary	rate	sing
1	Mark	Ballin	304	2327667	Anthony	AR	25813	S	Y	5000	3	200
8	Marcelino	Nuth	304	5404707	Kyle	WV	10000	M	N	10000	4	0

Supporting Rules Discovery

- Large literature on Functional Dependencies [Kivinen and Mannila, 1995]
- More recent efforts on data quality rules
  - Conditional Functional Dependencies [Chiang and Miller, 2008]
  - Matching Dependencies [Song and Chen, 2009]
  - Denial Constraints [Xu et al, 2013b]

Three Big Data challenges

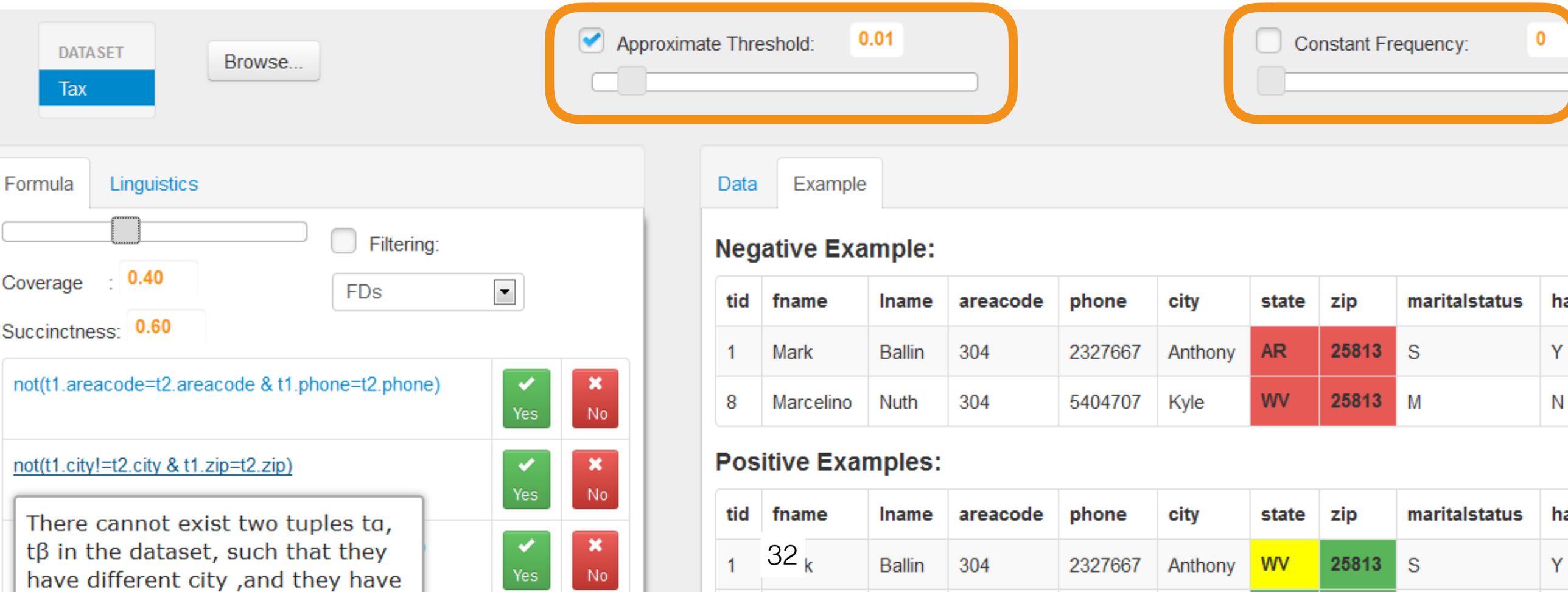
1. **Noise** in the data: hard to set parameters

2. Search space is **exponential**: no trial and error

3. **Lots** of rules, unfriendly output for domain experts

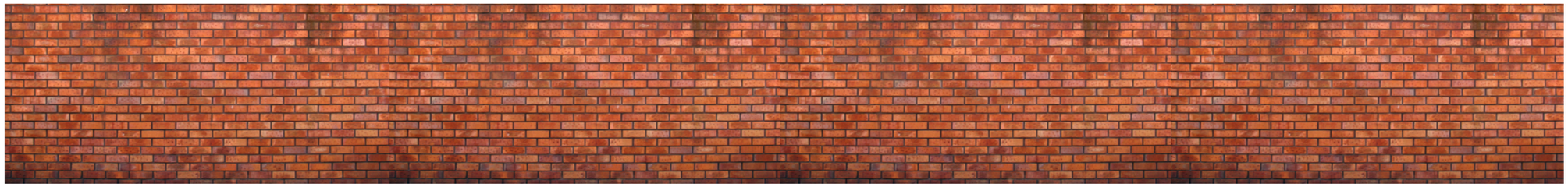
- Same problem for other methods in curation: transformations, outliers detection, deduplication

Rule Annotated Over Data	# Annotated Tuples	% Errors
acquired company → acquirer company	217	26
company → employees number	198	26
company → meeting type	179	17
ticker → company	1,906	4
company → new rank	150	8
person → company	186	14



New (ML/PL) tools to the rescue

- DCs cleaning and mining [Xu et al, 2013a] [Xu et al, 2013b]



- Temporal rules from **noisy** data [Abedjan et al, 2015]
- **Interactive** discovery with domain experts [He et al, 2016]
- Synthesizing cleaning **programs** (UDFs) [Singh et al, 2017]

Program synthesis

name	address	email	nation	gender
Catherine Zeta-Jones	9601 Wilshire Blvd., Beverly Hills, CA 90210-5213	c.jones@gmail.com	Wales	F
C. Zeta-Jones	3rd Floor, Beverly Hills, CA 90210	c.jones@gmail.com	US	F
Michael Jordan	676 North Michigan Avenue, Suite 293, Chicago		US	M
Bob Dylan	1230 Avenue of the Americas, NY 10020		US	M

name	apt	email	country	sex
Catherine Zeta-Jones	9601 Wilshire, 3rd Floor, Beverly Hills, CA 90210	c.jones@gmail.com	Wales	F
B. Dylan	1230 Avenue of the Americas, NY 10020	bob.dylan@gmail.com	US	M
Michael Jordan	427 Evans Hall #3860, Berkeley, CA 94720	jordan@cs.berkeley.edu	US	M

ML black box

Best F-measure
Not interpretable

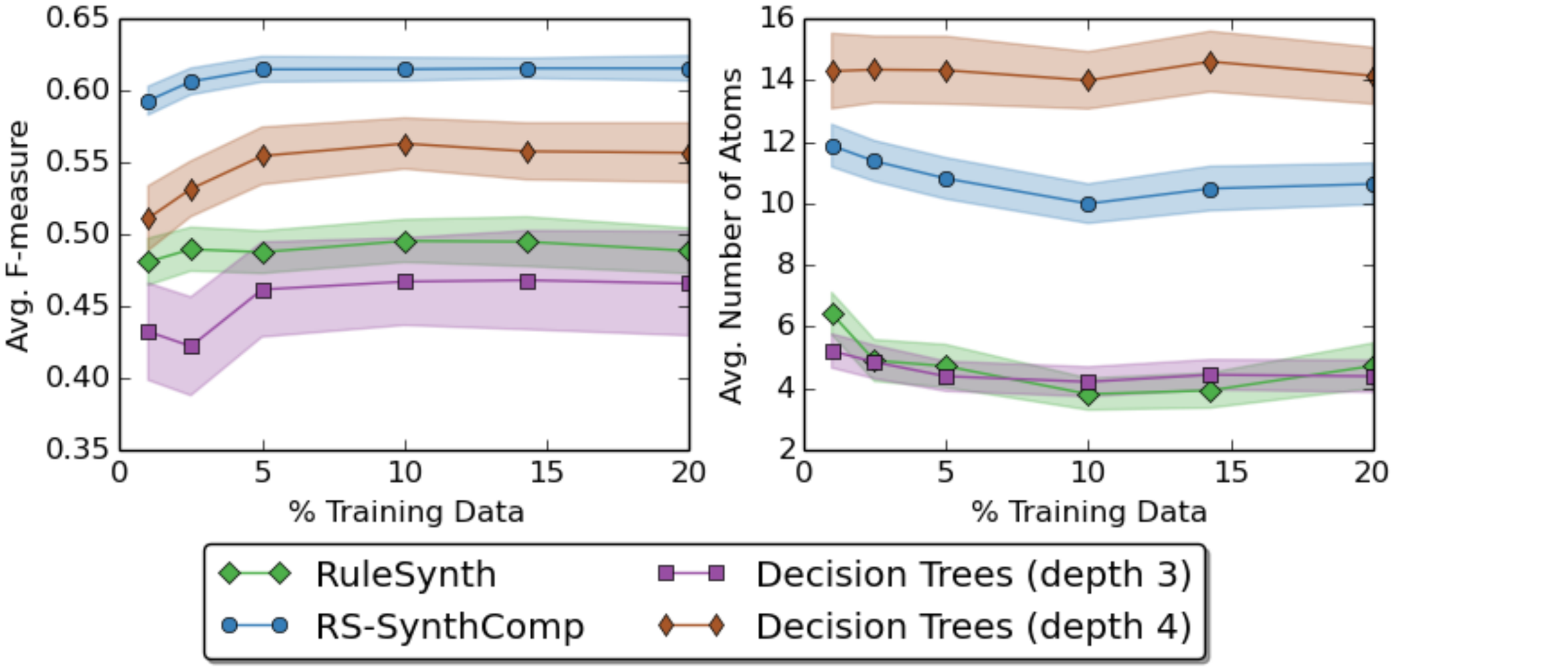
Rules

Lower F-measure
Interpretable

$$
\begin{array}{ll}
\textbf{if} & (r[\text{email}] \neq \text{Null} \bigwedge s[\text{email}] \neq \text{Null}) \\
\textbf{then} & r[\text{name}] \approx_1 s[\text{name}] \bigwedge r[\text{email}] = s[\text{email}] \\
\textbf{else} & r[\text{name}] \approx_3 s[\text{name}] \bigwedge r[\text{address}] \approx_2 s[\text{apt}] \bigwedge \\
 & r[\text{nation}] = s[\text{country}] \bigwedge r[\text{gender}] = s[\text{sex}]
\end{array}
$$

Tuneable trade off

Program synthesis



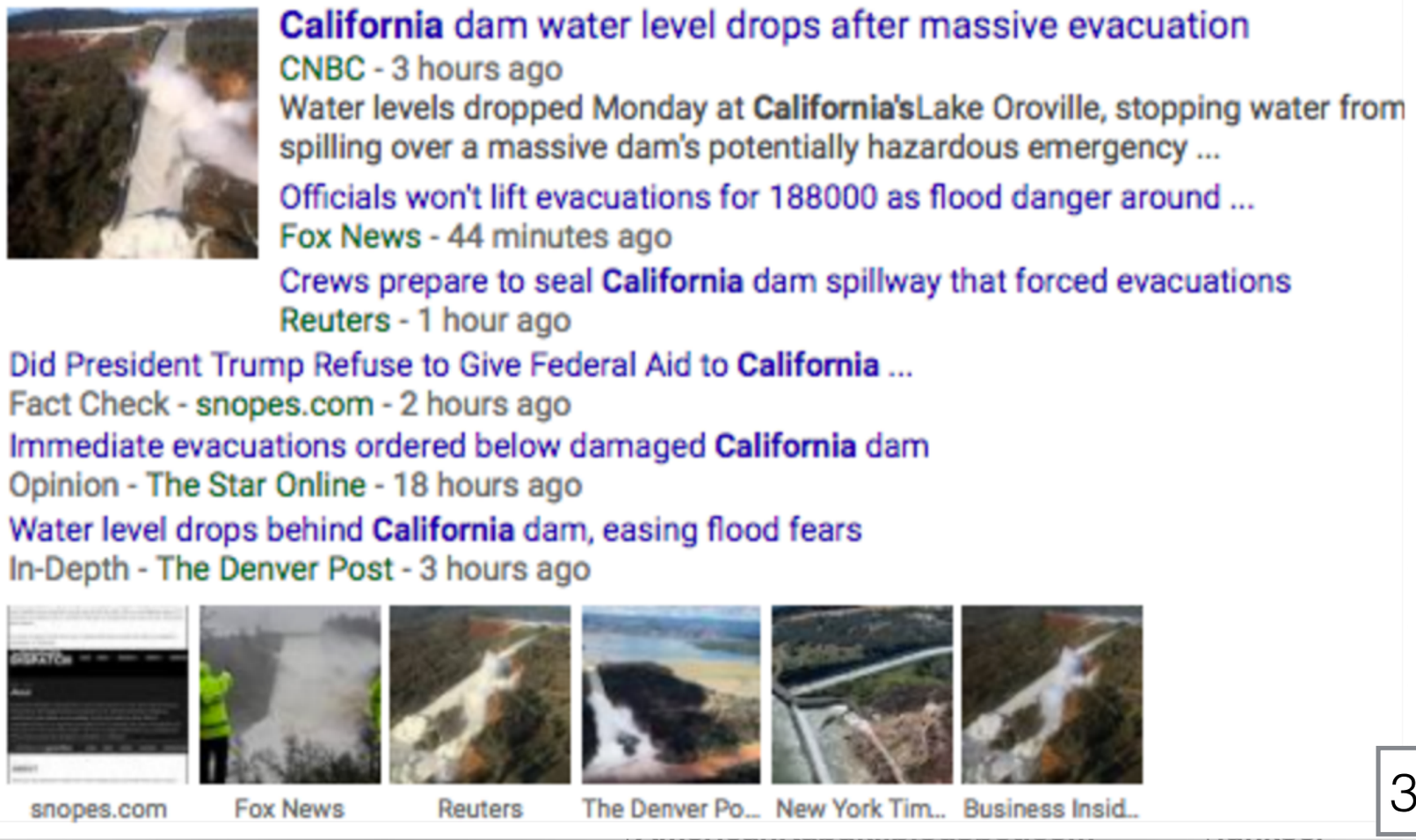
F-measure comparable to DTs depth 10 and SVM

Research Direction

- Rules for **challenging applications**
  - fact checking
  - identification of cyber attacks
  - recognizing credit card frauds

[www.opensources.co]
881 sources
"~200 suggested waiting to be added"

120 organizations, at least two days delay



POLITICS

Fact Check: Donald Trump's Speech On Immigration

August 31, 2016 · 9:44 PM ET


DOMENICO MONTANARO


DANIELLE KURTZLEBEN


SCOTT HORSLEY


SARAH MCCAMMON


RICHARD GONZALES

But these facts are never reported. Instead the media, and my opponent, discuss one thing and only one thing: The needs of people living here illegally. In many cases, by the way, they're treated better than our vets. Not going to happen anymore, folks, Nov. 8. Not going to happen anymore.

The truth is the central issue is not the needs of the 11 million illegal immigrants or however many there may be — and honestly we've been hearing that number for years. It's always 11 million. Our government has no idea. It could be 3 million, it could be 30 million, they have no idea what the number is.

> *[The count of immigrants in the country illegally is an estimate, but several estimates put it in the same ballpark — and it's the 11 million ballpark, nowhere near 30 million. The Pew Research Center puts it at 11.3 million (a number that has held relatively steady for years, by its estimate, and is down by nearly 1 million from a recent peak in 2007). As of January 2012, the Department of Homeland Security put the count at 11.4 million. Trump has in fact made the 30*

[http://www.npr.org/2016/08/31/492096565/fact-check-donald-trumps-speech-on-immigration]

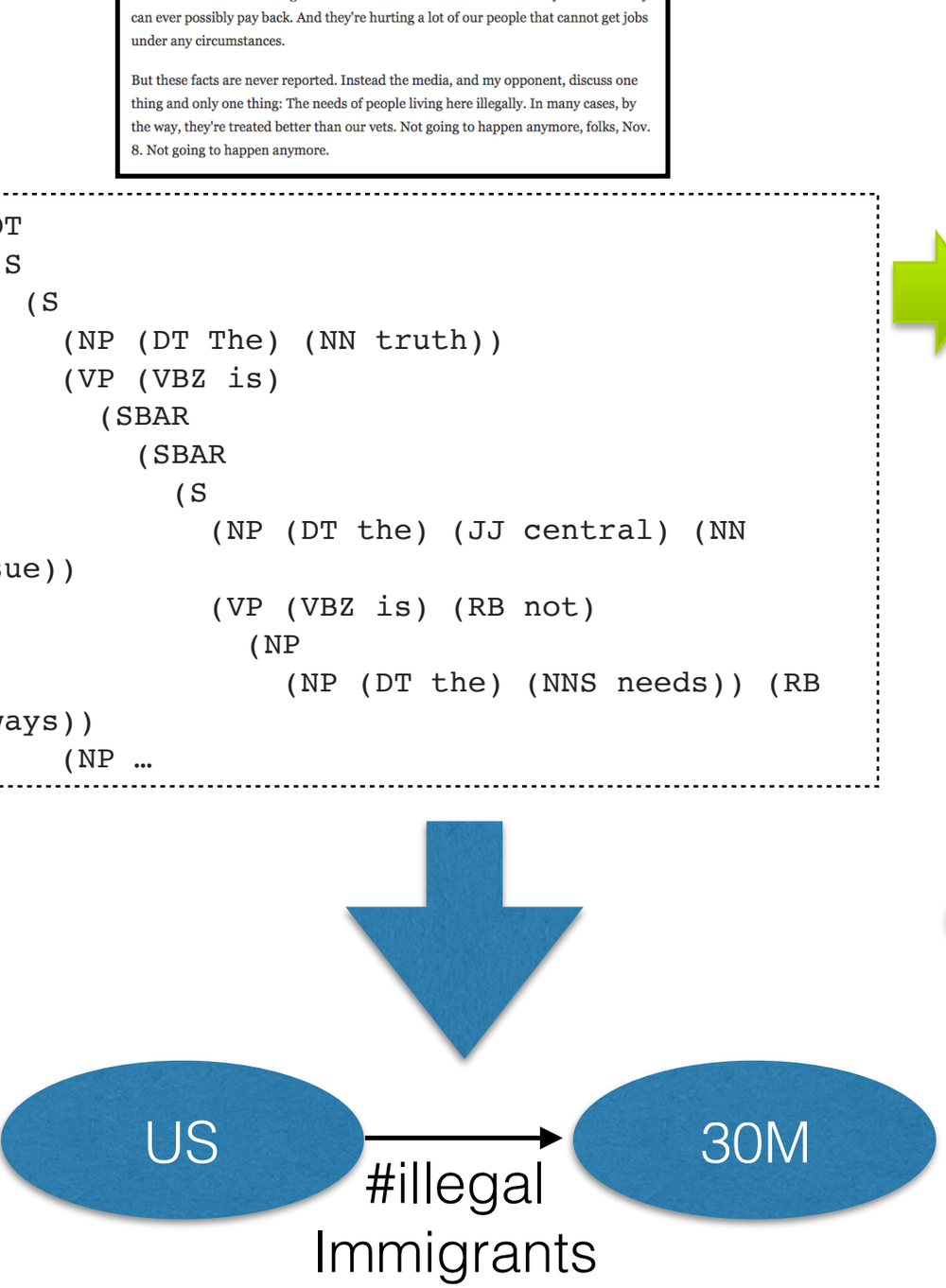
A 2011 report from the Government Accountability Office found that illegal immigrants and other non-citizens in our prisons and jails together had around 25,000 homicide arrests to their names. 25,000. On top of that, illegal immigration costs our country more than $113 billion a year, and this is what we get.

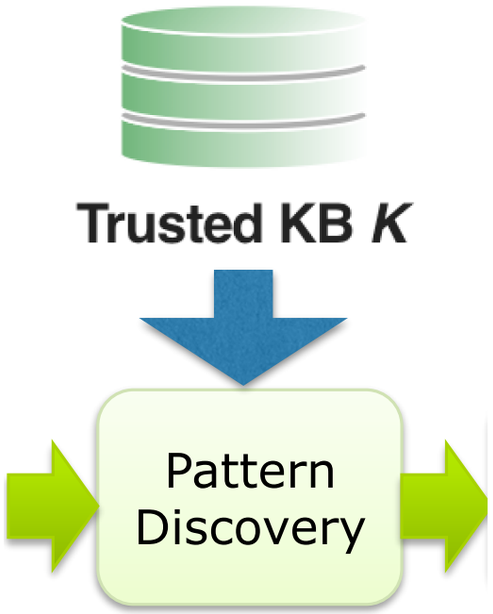
For the money we are going to spend on illegal immigration over the next 10 years, we could provide one million at-risk students with a school voucher, which so many people are wanting.

While there are many illegal immigrants in our country who are good people, many, many, this doesn't change the fact that most illegal immigrants are lower-skilled workers with less education who compete directly against vulnerable American workers and that these illegal workers draw much more out from the system than they can ever possibly pay back. And they're hurting a lot of our people that cannot get jobs under any circumstances.

But these facts are never reported. Instead the media, and my opponent, discuss one thing and only one thing: The needs of people living here illegally. In many cases, by the way, they're treated better than our vets. Not going to happen anymore, folks, Nov. 8. Not going to happen anymore.

```
ROOT
  (S
    (S
      (NP (DT The) (NN truth))
      (VP (VBZ is)
        (SBAR
          (SBAR
            (S
              (NP (DT the) (JJ central) (NN
issue))
              (VP (VBZ is) (RB not)
                (NP
                  (NP (DT the) (NNS needs)) (RB
always))
      (NP …
```



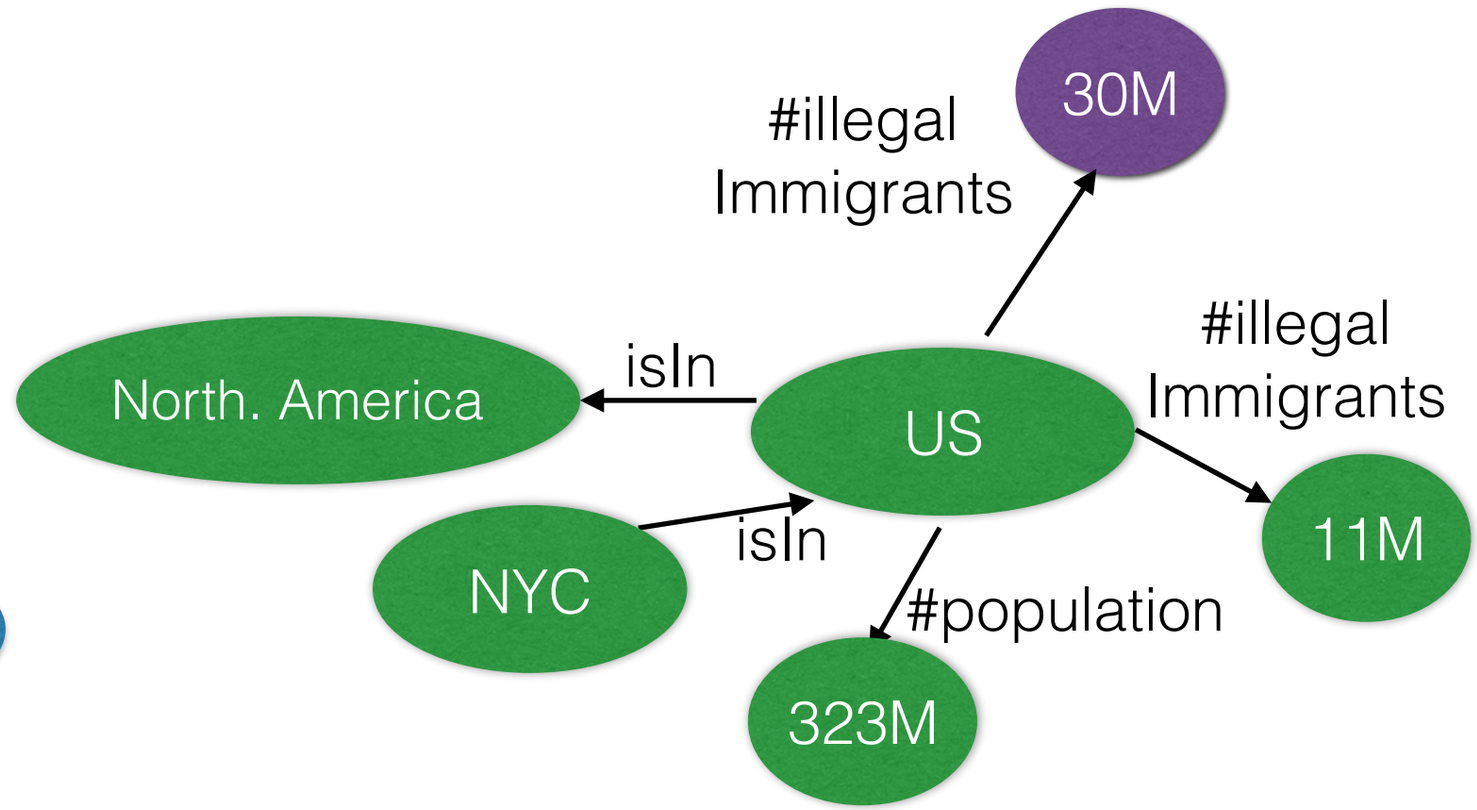


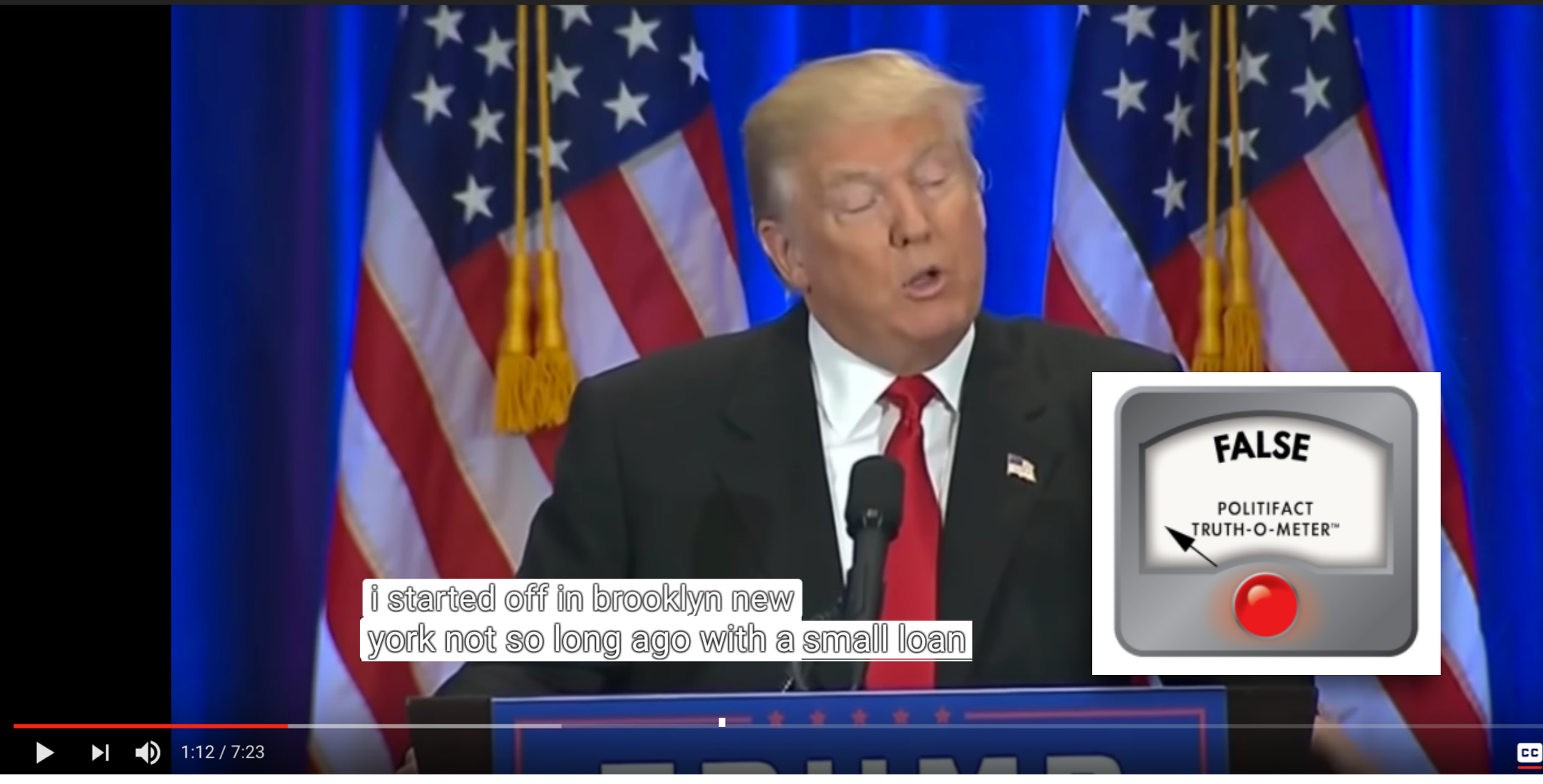
For the money we are going to spend on illegal immigration over the next 10 years, we could provide one million at-risk students with a school voucher, which so many people are wanting.

While there are many illegal immigrants in our country who are good people, many, many, this doesn't change the fact that most illegal immigrants are lower-skilled workers with less education who compete directly against vulnerable American workers and that these illegal workers draw much more out from the system than they can ever possibly pay back. And they're hurting a lot of our people that cannot get jobs under any circumstances.

But these facts are never reported. Instead the media, and my opponent, discuss one thing and only one thing: The needs of people living here illegally. In many cases, by the way, they're treated better than our vets. Not going to happen anymore, folks, Nov. 8. Not going to happen anymore.

The truth is the central issue is not the needs of the 11 million illegal immigrants or however many there may be — and honestly we've been hearing that number for years. It's always 11 million. Our government has no idea. It could be 3 million, it could be 30 million, they have no idea what the number is.





Make it explicable with **rules** over the KB!

Conclusions

- Big challenges in data cleaning
- No magic: large human involvement
- New tools for the existing problems
- New applications for the existing tools

Paolo Papotti
papotti@eurecom.fr
Gdansk, 11/9/2017