Active Learning for Large-Scale Entity Resolution

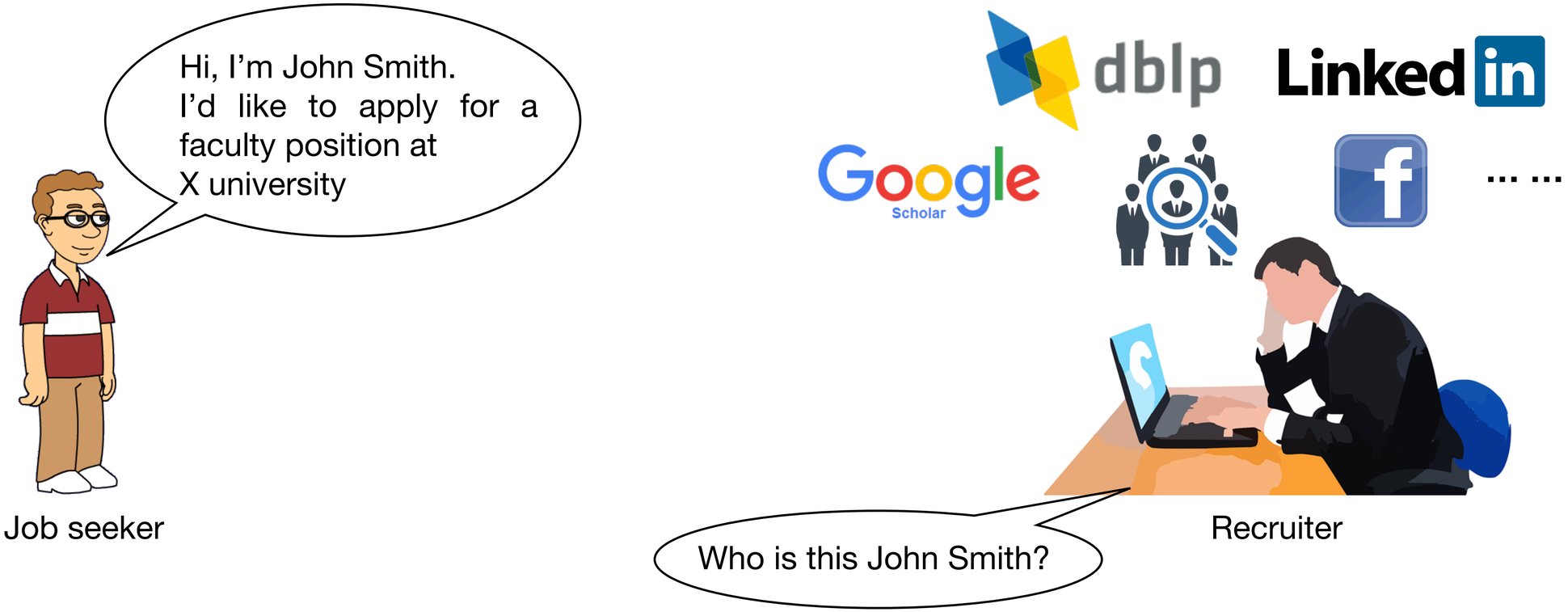
CIKM 2017 - Singapore

Kun Qian, Lucian Popa, Prithviraj Sen

IBM Research - Almaden

IBM

What is Entity Resolution (ER)?



- ER is the task of identifying and linking different representations of the same real world object.

  - An important task for many data analytic applications, such as knowledge base creation, information retrieval

What does ER in big data scenarios look like?

Social network data

Enterprise data

328 millions active users in Q1 2017

FROM TWITTER'S QUARTERLY PRESS RELEASES

Which IBM employees tweet?

About 380K employees

- **Challenges of large-scale entity resolution**
  - Scalability issue ***$O(10^{14})$ - cross product of Twitter-IBM***
  - Low matching ratio issue ***Hard to use supervised learning methods***
  - High human effort issue

Prior work

- Crowdsourcing can distribute labeling effort to the crowd with lower cost

  e.g., [Demartini et al. 2013, Khan 2016, Verroios et al. 2015, Vesdapunt et al. 2014, Wang et al. 2012, Whang et al. 2013]

- Active learning can reduce effort required by domain experts

  e.g., [Sarawagi & Bhamidipaty.2002, Arasu et al. 2010]

  - [Arasu et al. 2010] proposed a way to learn high-precision algorithm
    - Uses blocking functions to avoid computing cross product
    - *Falls short of producing high-recall results*

- **Achieving both high precision and high recall with low human effort is still a challenge**

We introduce **ERLearn**

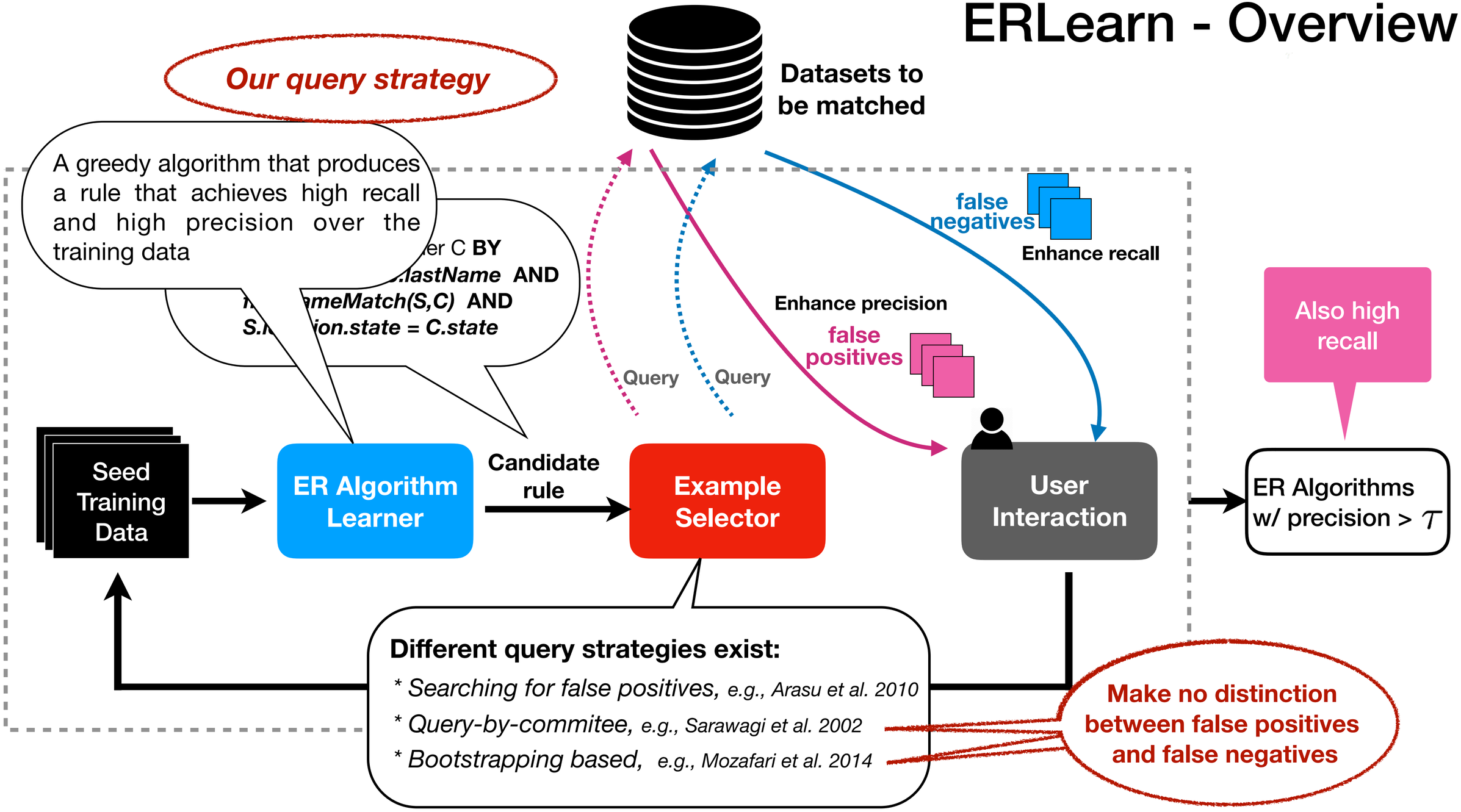
- **An active learning based system**
  - Requires only a small number of labels from one domain expert

- **Produces high precision and high recall ER algorithms at scale**
  - Learns under a precision constraint (e.g., >90%)
  - A new query strategy for finding informative examples
  - Efficient evaluation over large-scale datasets

Roadmap

- Motivation and Introduction
- Main Features of ERLearn
- **Overview of ERLearn**
- Details of Query Strategy
- Experiments
- Conclusion

ERLearn - Overview



An Example Run of ERLearn

Seed training data

Social network profile

Screen_name: @locke1
Name: Johnny Locke
Location: OH

...

Customer records

ID	Name	City	State	Street	Email
23	John Locke	Dayton	OH	1174 Hill Rock Way	jl@c.com

...

Initial rule learned from seed training data (may be *inaccurate*)

ER ... BY

Example Scenario

Social network profiles vs. customer records

...rule after ...ling of new examples

Example Selector

Social network profile

Screen_name: @rgreen
Name: Bob Green
Location: Los Angeles, CA

Customer records

ID	Name	City	State	Street	Email
8	Robert Green	LA	N/A	…	rgreen@c.com

False negative

Match social S, customer C **BY**
S.lastName = C.lastName **AND**
firstNameMatch(S.firstname, C.firstname) **AND**
S.screen_name = C.emailHandle

Social network profile

Screen_name: @emms7
Name: Emily Waters
Location: Sugar Land, TX

...

Customer records

ID	Name	City	State	Street	Email
7	Emily Waters	El Paso	TX	23 Sims Dr Apt 2	ew@c.com

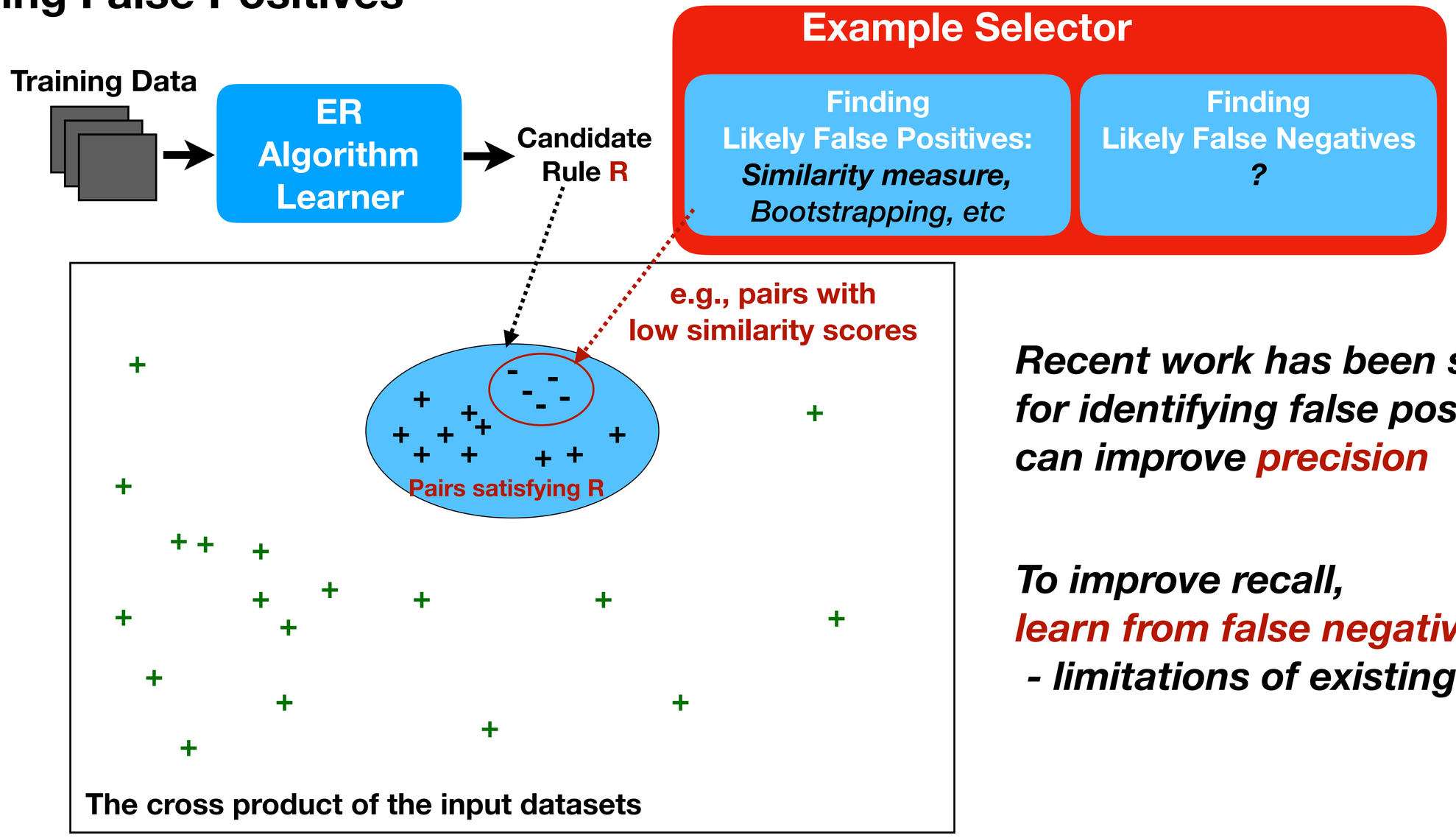
...

False positive

Roadmap

- Motivation and Introduction
- Main Features of ERLearn
- Overview of ERLearn
- **Details of Query Strategy**
  - **Finding false positives and false negatives**
- Experiments
- Conclusion

Finding False Positives



Recent work has been successful for identifying false positives, which can improve precision

To improve recall,
learn from false negatives!
- limitations of existing offerings

The search space of false negatives is extremely large (i.e., cross product) !!!

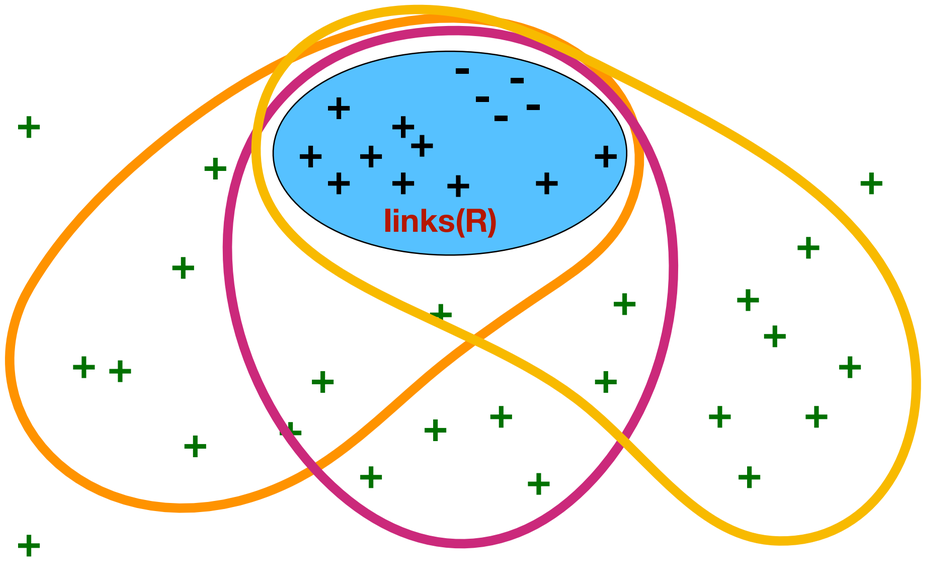
Example Selector - continue

Example Selector

Finding Likely False Positive: *Similarity measure, Bootstrapping, etc*	Finding Likely False Negatives *Rule-Minus*

***J* controls the jump size**

Rule-minus rules: relax a candidate rule **R** by removing *j* matching functions ***J* = 1, 2, …**



Find pairs with high similarity scores within the search spaces created by rule-minus rules

The cross product of the input datasets

Candidate rule R

S.firstname = C.first AND
S.city = C.city AND
S.lastname = c.last

***J* = 1**

~~S.firstname = C.first AND~~
S.city = C.city AND
S.lastname = c.last

S.firstname = C.first AND
~~S.city = C.city AND~~
S.lastname = c.last

S.firstname = C.first AND
S.city = C.city AND
~~S.lastname = c.last~~

Roadmap

- Motivation and Introduction
- Main Features of ERLearn
- Overview of ERLearn
- Details of Query Strategy
- **Experiments**
- Conclusion

Experimental Evaluation

- **Real-world matching scenarios:**
  - **DBLP-Scholar** - DBLP (~2.6K) vs. Google Scholar (64K) - *with ground truth*
  - **Social-Enterprise** - Social Network (50 million) vs. Enterprise customer data (~470K)

A real world big data scenario

- **Baselines**
  - *Active learning based*:
    - ALGPR [proposed by Arasu et al.],
    - ALIAS [proposed by Sarawagi et al.]
  - *Supervised learning:*
    - SVM
    - MLN (Markov Logic Network) [proposed by Singla and Domingos]

Experimental Evaluation

- **Compare with active learning approaches**
  - ALGPR (focusing on finding false positives)
  - ALIAS (Query-by-committee)

DBLP-Scholar Dataset (has ground truth)

Method	# Rules	Labels	Recall	Precision	F-score
ALIAS	n/a	**160**	0.82	0.75	0.78
ALGPR	2	210	0.67	**0.97**	0.80
ERLearn	**6**	163	**0.84**	0.90	**0.87**

Suboptimal

Highest F-score

Comparable to the minimum labels

Experimental Evaluation

- **Compare with more baselines in real world big data scenario**
  - ALGPR (focusing on finding false positives)
  - ALIAS (Query-by-committee)
  - SVM
  - Markov Logic Network (MLN)

Social-Enterprise Dataset

Method		Labels	Links	Precision	α
SVM		430	**21**	0.952	0.049
MLN		430	**84**	0.90	0.195
ALIAS		241	340	0.40	0.56
ALGPR		159	**181**	0.933	1.14
ERLEARN	$j = 1$	172	**683**	0.933	3.97
	$j = 2$	430	**1088**	0.933	2.533

α - # of true links generated per label

Suboptimal

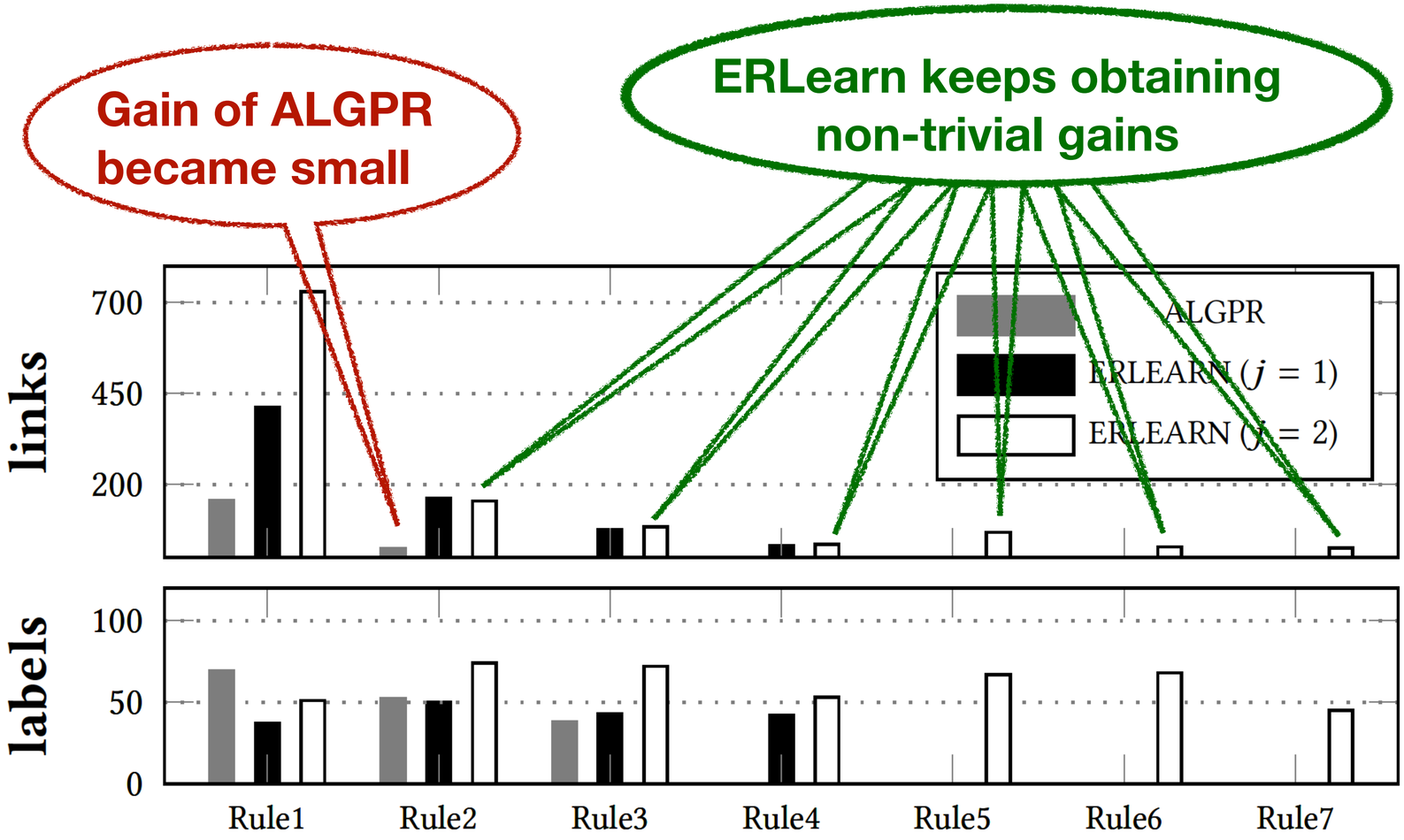
largest number of true links found

Highest effectiveness

IBM

Experimental Evaluation - vs. ALGPR

- **A zoom-in comparison of ALGPR and ERLearn**
  - ALGPR (focusing on finding ***false positives***)



- ***# new true matches found***
- ***# user labels***

Real Rules Learned by ERLearn

Pool of user-provide matching functions

- Equality function (x = y)
  - *Applicable to different attributes (e.g., city, states, country, lastname, firstname, emailhandle..*
- Textual similarity based functions
  - *Applicable to different attributes (e.g., city, states, country, lastname, firstname, emailhandle..*
- lastNameFrequencyFilter(x, *threshold*)
- firstNameMatch(x, y)
- countryInUSA(x)
- GeoLocationWithin (x, y, dist)
- … …. … … …

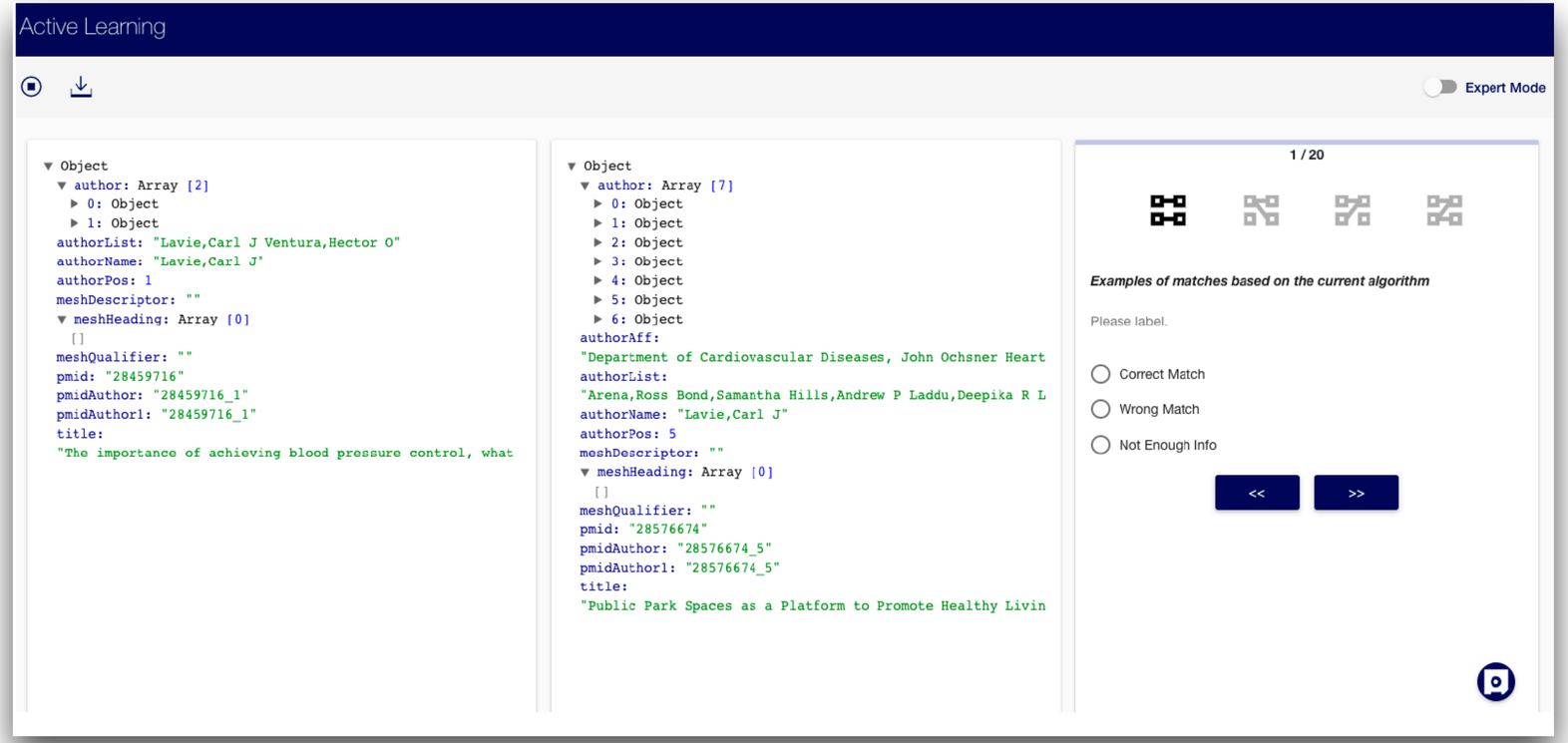
Match social s, enterprise i BY
upperCase(s.home.city) = upperCase(i.city)
AND s.screen_name = i.emailHandler
AND s.Name.last = i.name.last
AND lastNameFrequencyFilter(i.last, 85)

Match social s, enterprise i BY
firstNameMatch(i.Name.firstNameVars, s.name.first)
AND i.Name.last = s.name.last
AND lastNameFrequencyFilter (s.name.last, 60)
AND upperCase(i.CITY) = upperCase(s.home.city)
AND coutryIsInUSA(i.COUNTRY)

IBM

Current Status of ERLearn

- **Implementation**
  - ERLearn is web application with a spark back-end.



User Interface of ERLearn

- **Efficiency**
  - For Social-Enterprise Scenario, less than 1 minute to execute the candidate rule and to find examples to be labeled.

Roadmap

- Motivation and Introduction
- Main Features of ERLearn
- Overview of ERLearn
- Details of Query Strategy
- Experiments
- **Conclusion**

Conclusion

- **We introduced ERLearn, an active learning based system for large scale ER**

- **Experiments demonstrated that ERLearn can produce high-quality ER rules**
  - A new query strategy for finding false positives and false negatives

- **ERLearn requires a small amount of human effort**

- **Directions for future work**
  - Run it on more datasets
  - Learn matching functions

Thank you