Knowledge Extraction and Inference from Text: Shallow, Deep, and Everything in Between (Tutorial at SIGKDD 2018)

Partha Talukdar
IISc Bangalore and Kenome
ppt@iisc.ac.in

Tutorial homepage: https://goo.gl/vRkwxZ

August 19, 2018

Many slides reused from CIKM 2017 tutorial with Soumen Chakrabarti (IIT Bombay)

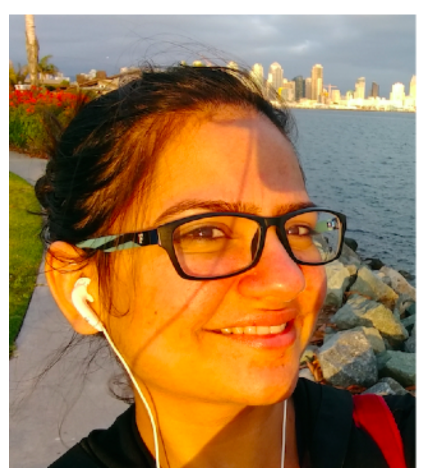
Acknowledgment

- Soumen Chakrabarti (IIT Bombay)
- Tom Mitchell (CMU)
- Masuam (IIT Delhi)

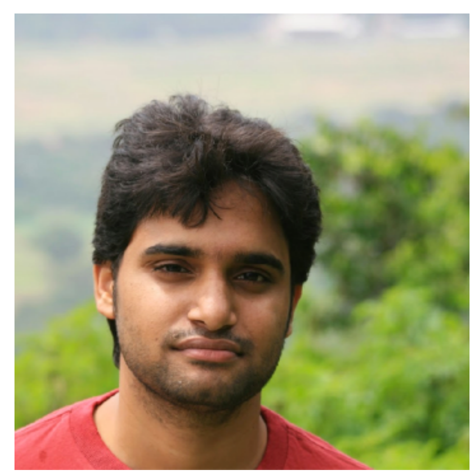
Acknowledgment

- Soumen Chakrabarti (IIT Bombay)
- Tom Mitchell (CMU)
- Masuam (IIT Delhi)
- My PhD students at MALL Lab, IISc Bangalore

Chandrahas Dewangan



Sharmistha Jat



Madhav Nimishakavi

Shikhar Vashishth

Explosion of Unstructured Text Data

Explosion of Unstructured Text Data

300 million new websites added in 2011 alone (a 117% growth)

Sources: http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/, http://blog.twitter.com/2011/06/200-million-tweets-per-day.html, https://commons.wikimedia.org/, https://pixabay.com/

Explosion of Unstructured Text Data

300 million new websites added in 2011 alone (a 117% growth)

500 million Tweets per day (circa Oct 2012)

Time to read for one person: 31years

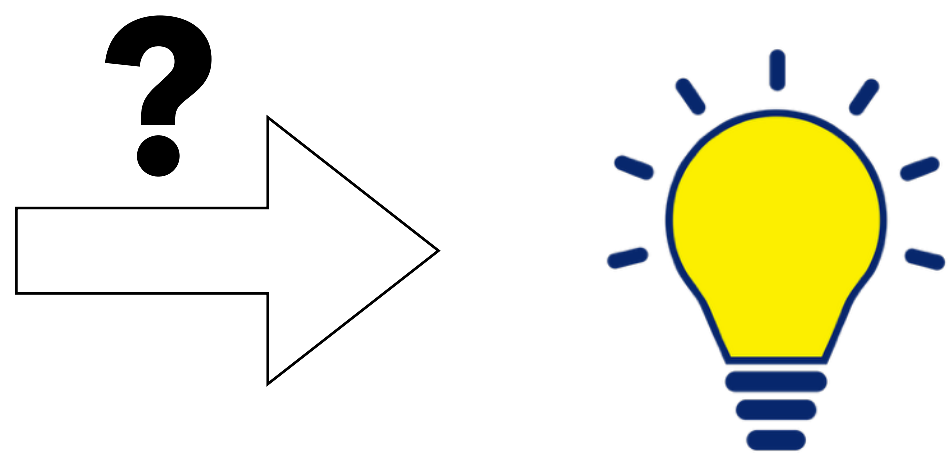
Sources: http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/, http://blog.twitter.com/2011/06/200-million-tweets-per-day.html, https://commons.wikimedia.org/, https://pixabay.com/

Explosion of Unstructured Text Data

300 million new websites added in 2011 alone (a 117% growth)

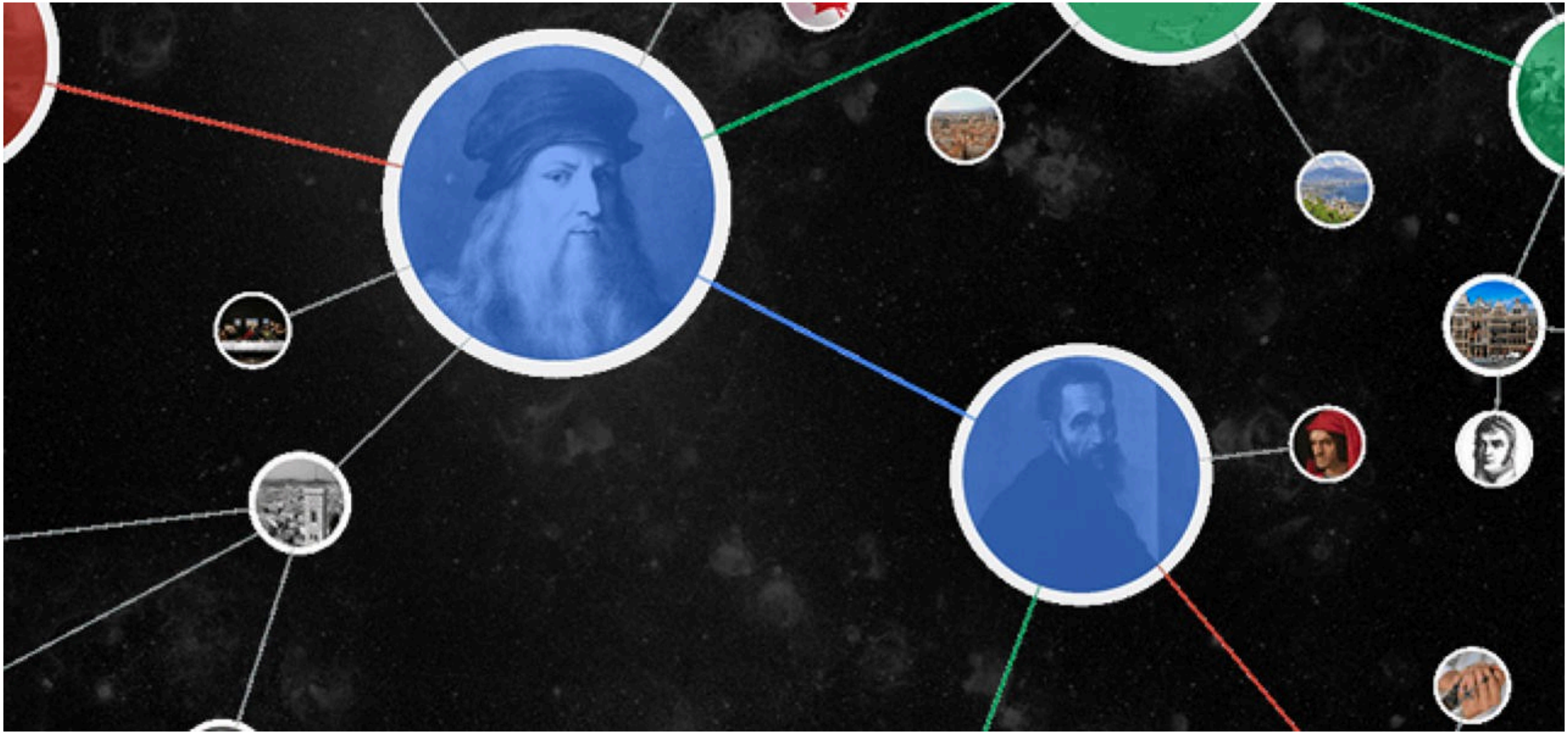
500 million Tweets per day (circa Oct 2012)

Time to read for one person: 31years



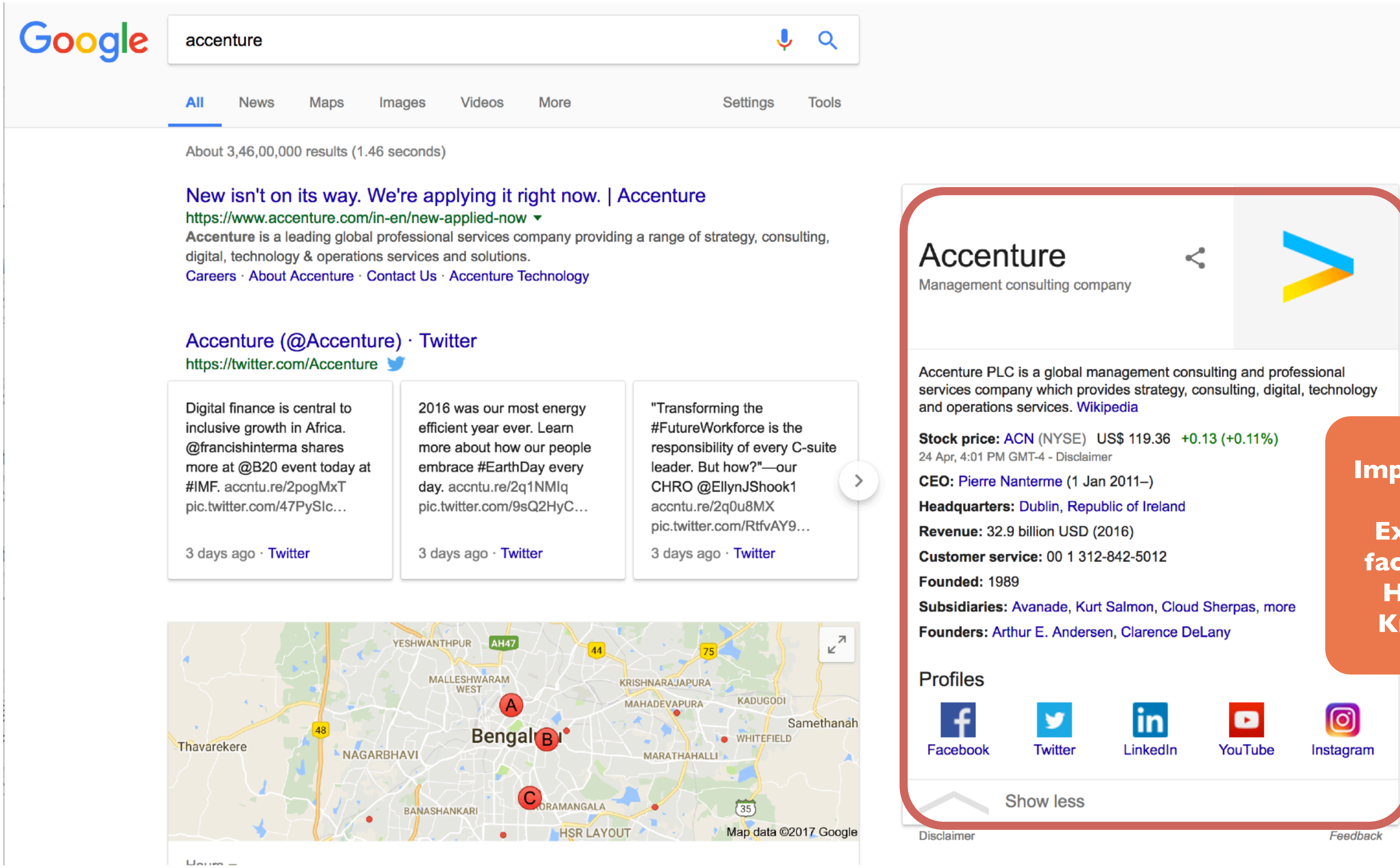
Sources: http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/, http://blog.twitter.com/2011/06/200-million-tweets-per-day.html, https://commons.wikimedia.org/, https://pixabay.com/

Knowledge Graph: Things, not Strings



Use case: Google Knowledge Graph

Use case: Google Knowledge Graph



Improved Web Search Experience, facilitated by Harvested Knowledge

Use case: GeoDeepDive and PaleoDeepDive

DeepDive builds KG out of scientific publications in Geology and Paleontology domains

Use case: Conversational AI

Use case: Conversational AI

Use case: Conversational AI

Knowledge Graphs can provide a shared context

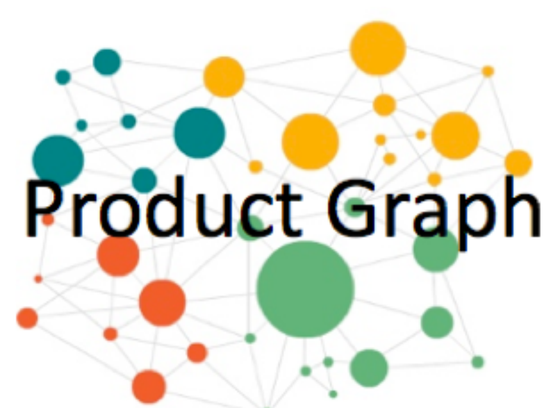
Use case: Conversational AI

Knowledge Graphs can provide a shared context

Google
Knowledge Graph

Facebook Entity Graph

Amazon



Microsoft Satori

LinkedIn Graph

Tutorial Focus

Weakly-supervised methods for Knowledge Graph (KG) construction

For additional topics on inference over KG, typing, entity linking, etc.,

please see SIGIR 2018 tutorial slides at https://goo.gl/vRkwxZ

Outline

13:00-13:15 Overview and motivation

13:15-13:45 Case study: NELL

13:45-14:00 Bootstrapped Entity Extraction

14:00-15:00 Open Relation Extraction & Canonicalization

15:00-15:30 **Coffee Break**

15:30-16:15 Distantly-supervised Neural Relation Extraction

16:15-16:45 Knowledge Graph Embeddings

16:45-17:00 Conclusion & QA

Outline

13:00-13:15 Overview and motivation

13:15-13:45 Case study: NELL

13:45-14:00 Bootstrapped Entity Extraction

14:00-15:00 Open Relation Extraction & Canonicalization

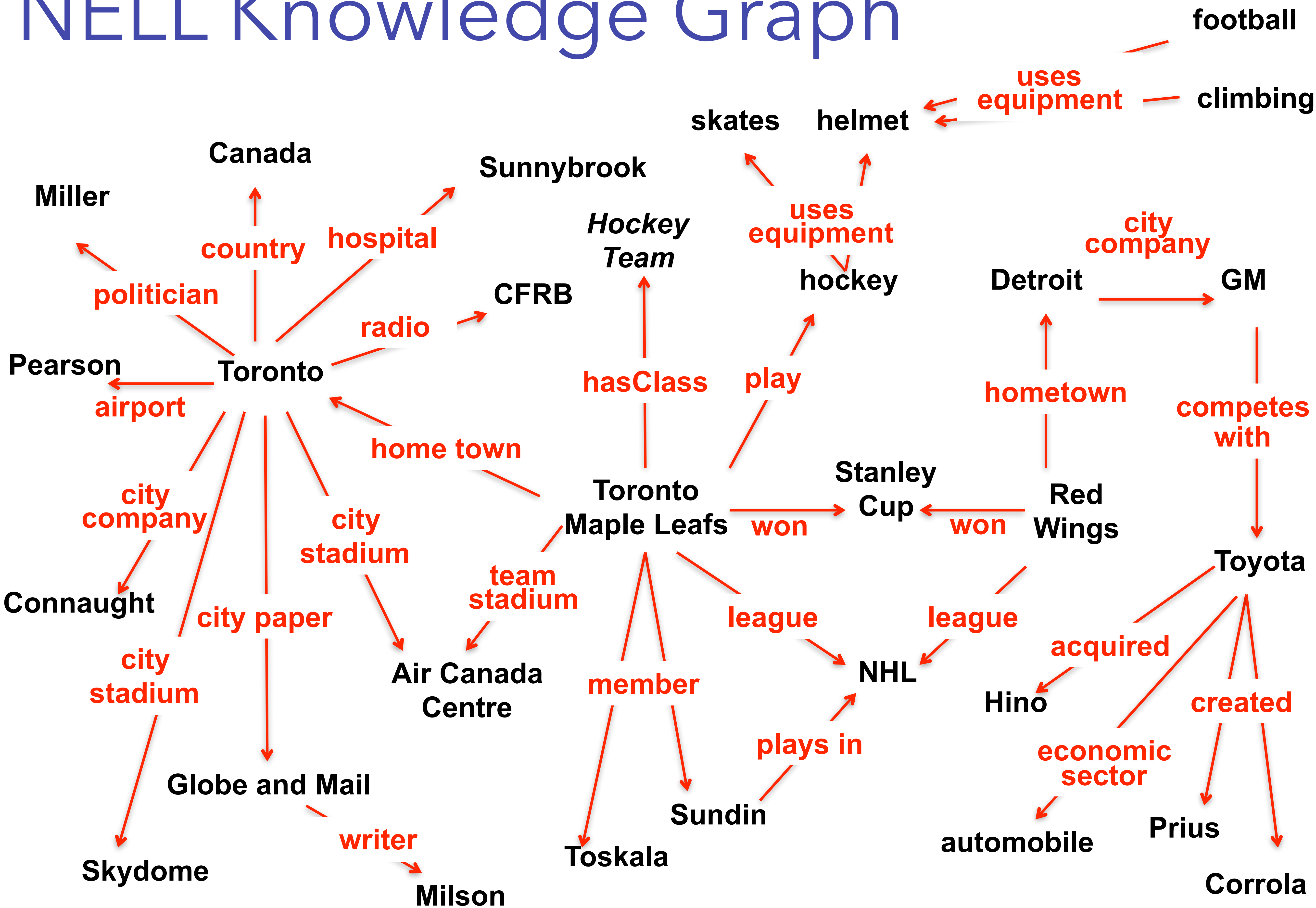
15:00-15:30 **Coffee Break**

15:30-16:15 Distantly-supervised Relation Extraction

16:15-16:45 Knowledge Graph Embeddings

16:45-17:00 Conclusion & QA

NELL Knowledge Graph



New paradigm for Machine Learning:

Never Ending Learning agent

New paradigm for Machine Learning:

Never Ending Learning agent

Persistent software individual

New paradigm for Machine Learning:

Never Ending Learning agent

Persistent software individual

Learns many functions / knowledge types

New paradigm for Machine Learning:

Never Ending Learning agent

Persistent software individual

Learns many functions / knowledge types

Learns easier things first, then more difficult

New paradigm for Machine Learning:

Never Ending Learning agent

Persistent software individual

Learns many functions / knowledge types

Learns easier things first, then more difficult

The more it learns, the more it can learn next

New paradigm for Machine Learning:

Never Ending Learning agent

Persistent software individual

Learns many functions / knowledge types

Learns easier things first, then more difficult

The more it learns, the more it can learn next

Learns from experience, and from advice

NELL: Never Ending Language Learner @ CMU

NELL: Never Ending Language Learner @ CMU

Inputs:

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate
- the web

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate
- the web
- occasional interaction with human trainers

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate
- the web
- occasional interaction with human trainers

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate
- the web
- occasional interaction with human trainers

The task:

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate
- the web
- occasional interaction with human trainers

The task:

- run 24x7, forever

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate
- the web
- occasional interaction with human trainers

The task:

- run 24x7, forever
- each day:

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate
- the web
- occasional interaction with human trainers

The task:

- run 24x7, forever
- each day:
    - extract more facts from the web

NELL: Never Ending Language Learner @ CMU

Inputs:

- initial ontology
- few seed examples of each ontology predicate
- the web
- occasional interaction with human trainers

The task:

- run 24x7, forever
- each day:
    - extract more facts from the web
    - learn to read (perform #1) better than yesterday

NELL Today

NELL Today

Running 24x7, since January, 12, 2010

Result:

KB with > 100 million candidate beliefs, growing daily

learning to reason, as well as read

automatically extending its ontology

NELL Today

Running 24x7, since January, 12, 2010

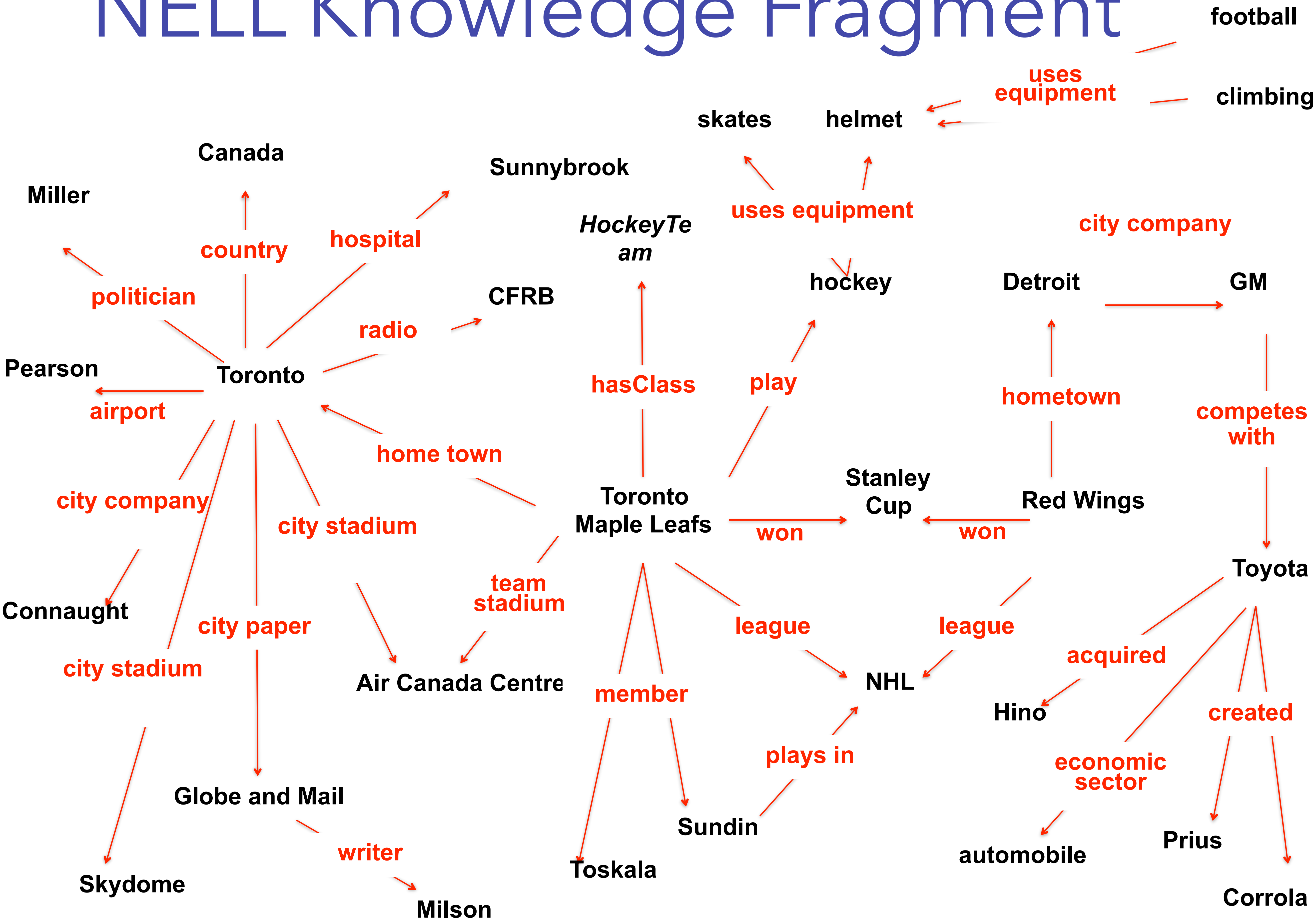
Result:

KB with > 100 million candidate beliefs, growing daily

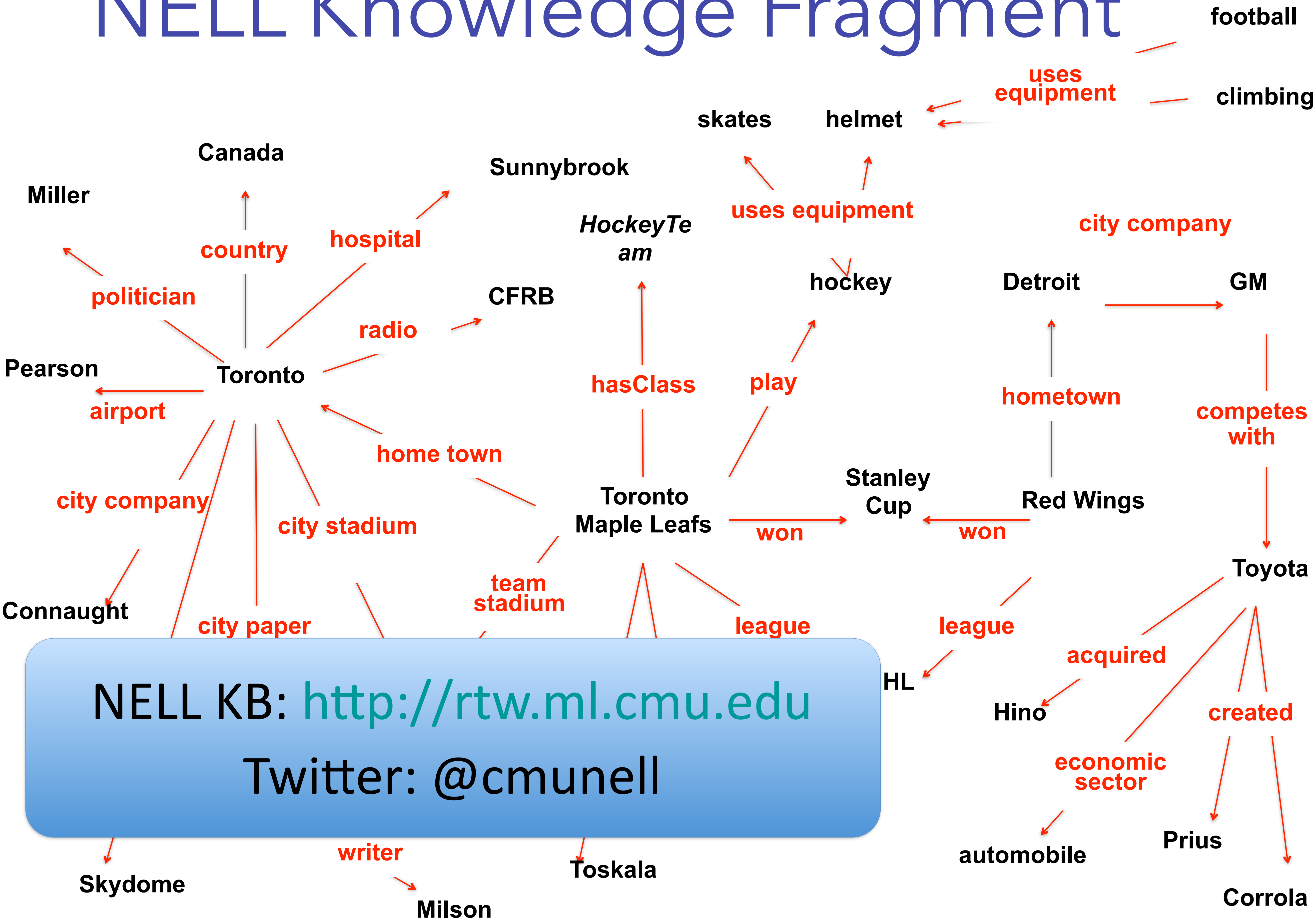
learning to reason, as well as read

automatically extending its ontology

NELL Knowledge Fragment



NELL Knowledge Fragment



NELL KB: http://rtw.ml.cmu.edu

Twitter: @cmunell

NELL Today

- eg. "diabetes", "Avandia", "tea", "IBM", "love" "baseball" "BacteriaCausesCondition" "kitchenItem" "ClothingGoesWithClothing" …

Recently-Learned Facts twitter

instance	iteration	date learned
mark_bellhorn is a Mexican person	763	27-aug-2013
methenamine_mandelate_tablet is a drug	763	27-aug-2013
pete_zimmer is a person	763	27-aug-2013
sandhills_clubtail is a vertebrate	764	31-aug-2013
jeffrey_carlson is a chef	763	27-aug-2013
sutton is a park in the city london	767	06-sep-2013
pushkin was born in moscow	767	06-sep-2013
honda is a company that produces accord	766	04-sep-2013
spurs is a sports team that plays against magic	763	27-aug-2013
baseball is a sport played in the venue ballpark_in_arlington	766	04-sep-2013

Other Related Efforts

Cycorp®
Home of smarter solutions

Freebase

yago
select knowledge

Google
"KNOWLEDGE VAULTS"

NELL

High Supervision

Low Supervision

DOI:0 0.0000/000

Never-Ending Learning

By T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling

Abstract

Whereas people learn many different types of knowledge from diverse experiences over many years, and become better learners over time, most current machine learning systems are much more narrow, learning just a single function or data model based on statistical analysis of a single data set. We suggest that people learn better than computers precisely because of this difference, and we suggest a key direction for machine learning research is to develop software architectures that enable intelligent agents to also learn many types of knowledge, continuously over many years, and to become better learners over time. In this paper we define more precisely this *never-ending learning* paradigm for machine learning, and we present one case study: the Never-Ending Language Learner (NELL), which achieves a number of the desired properties of a never-ending learner. NELL has been learning to read the Web 24hrs/day since January 2010, and so far has acquired a knowledge base with 120mn diverse, confidence-weighted beliefs (e.g., *servedWith(tea,biscuits)*), while learning thousands of interrelated functions that continually improve its reading competence over time. NELL has also learned to reason over its knowledge base to infer new beliefs it has not yet read from those it has, and NELL is inventing new relational predicates to extend the ontology it uses to represent beliefs. We describe the design of NELL, experimental results illustrating its behavior, and discuss both its successes and shortcomings as a case study in never-ending learning. NELL can be tracked online at http://rtw.ml.cmu.edu, and followed on Twitter at @CMUNELL.

1. INTRODUCTION

Machine learning is a highly successful branch of Artificial Intelligence (AI), and is now widely used for tasks from spam filtering, to speech recognition, to credit card fraud detection, to face recognition. Despite these successes, the ways in which computers learn today remain surprisingly narrow when compared to human learning. This paper explores an alternative paradigm for machine learning that more closely models the diversity, competence and cumulative nature of human learning. We call this alternative paradigm *never-ending learning*.

To illustrate, note that in each of the above machine learning applications, the computer learns only a single function to perform a single task in isolation, usually from human labeled training examples of inputs and outputs of that function. In spam filtering, for instance, training examples consist of specific emails and spam or not-spam labels for each. This style of learning is often called *supervised function approximation*, because the abstract learning problem is to approximate some unknown function $f: X \rightarrow Y$ (e.g., the spam filter) given a training set of input/output pairs $\{\langle x_i, y_i \rangle\}$ of that function. Other machine learning paradigms exist as well (e.g., unsupervised clustering, topic modeling, reinforcement learning) but these paradigms also typically acquire only a single function or data model from a single dataset.

In contrast to these paradigms for learning single functions from well organized data sets over short time-frames, humans learn many different functions (i.e., different types of knowledge) over years of accumulated diverse experience, using extensive background knowledge learned from earlier experiences to guide subsequent learning. For example, humans first learn to crawl, then to walk, run, and perhaps ride a bike. They also learn to recognize objects, to predict their motions in different circumstances, and to control those motions. Importantly, they learn *cumulatively*: as they learn one thing this new knowledge helps them to more effectively learn the next, and if they revise their beliefs about the first then this change refines the second.

The thesis of our research is that *we will never truly understand machine or human learning until we can build computer programs that, like people,*

- learn many different types of knowledge or functions,
- from years of diverse, mostly self-supervised experience,
- in a staged curricular fashion, where previously learned knowledge enables learning further types of knowledge,
- where self-reflection and the ability to formulate new representations and new learning tasks enable the learner to avoid stagnation and performance plateaus.

We refer to this learning paradigm as "never-ending learning." The contributions of this paper are to (1) define more precisely the never-ending learning paradigm, (2) present as a case study a computer program called the NELL which implements several of these capabilities, and which has been learning to read the Web 24hrs/day since January 2010, and (3) identify from NELL's strengths and weaknesses a number of key design features important to any never-ending learning system. This paper is an elaboration and extension to an earlier overview of the NELL system.[27]

2. RELATED WORK

Previous research has considered the problem of designing machine learning agents that persist over long periods

The original version of this paper appeared in the *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (Austin, TX, Jan. 25–30, 2015), 2302–2310.

CACM 2018

NELL's Growth over Time

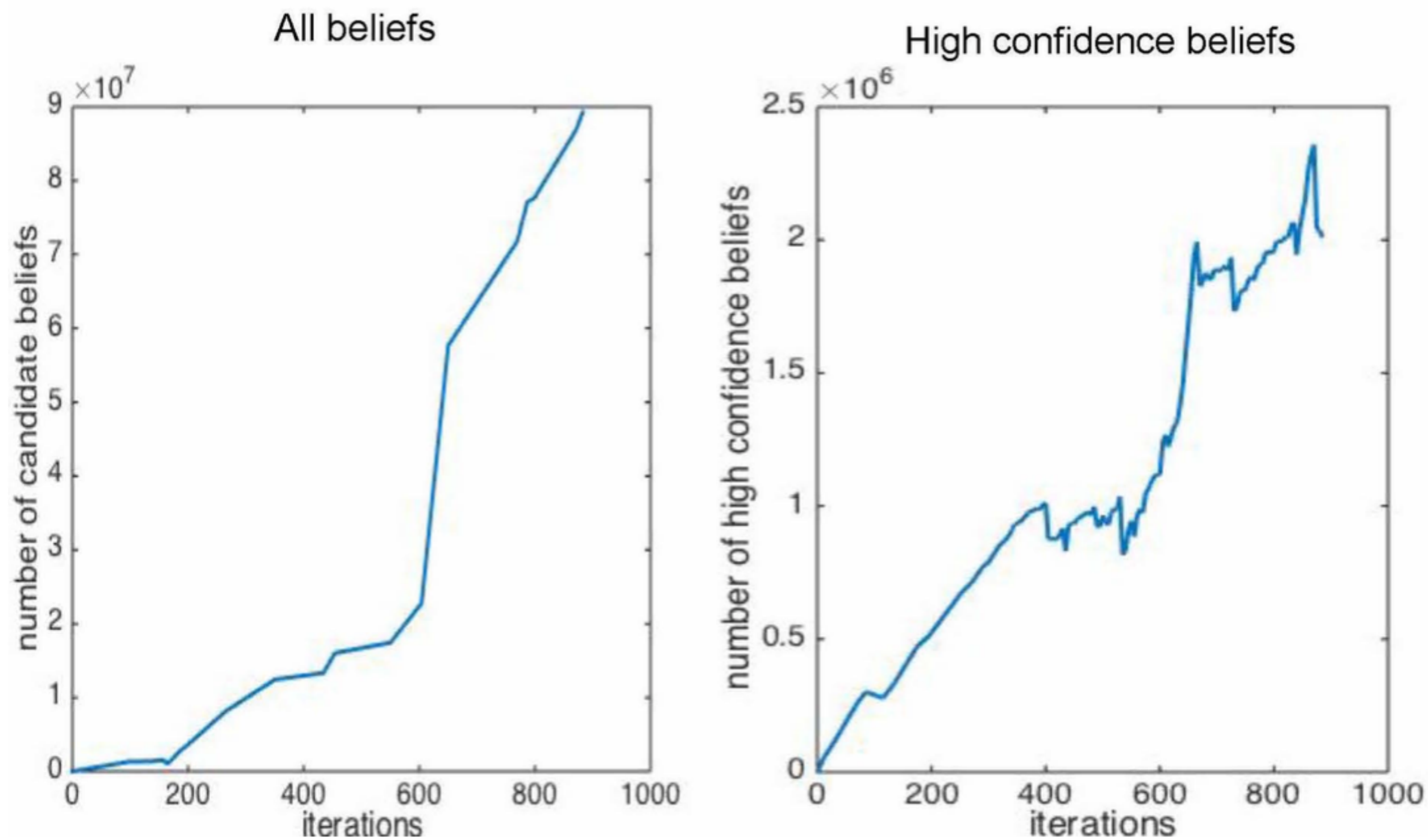
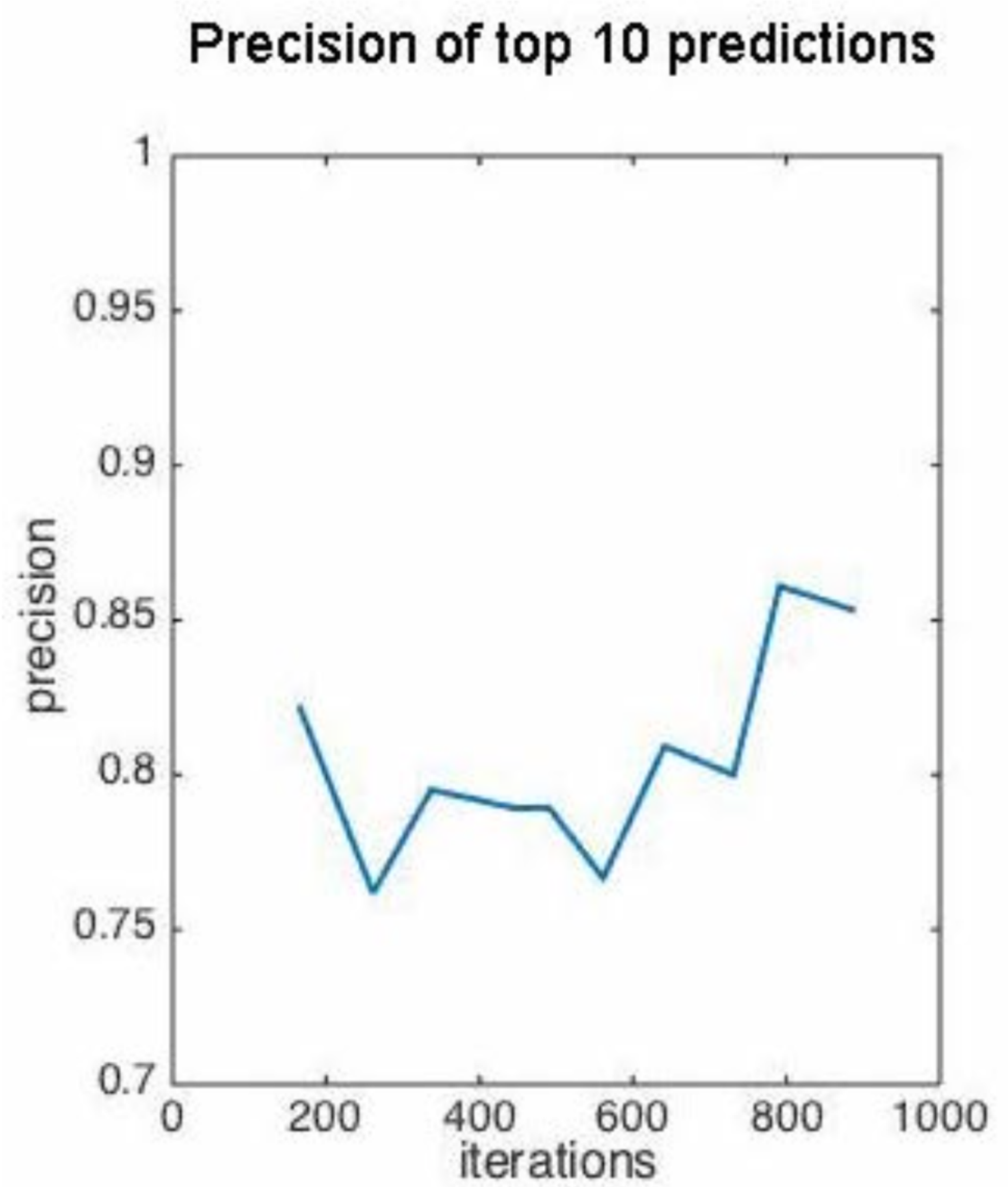


Figure 3: **NELL KB size over time.** Total number of beliefs (left) and number of high confidence beliefs (right) versus iterations. Left plot vertical axis is tens of millions, right plot vertical axis is in millions.

NELL's Accuracy over Time



How does NELL work?

Semi-Supervised Bootstrap Learning

Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

Semi-Supervised Bootstrap Learning

Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

mayor of arg1
live in arg1

Semi-Supervised Bootstrap Learning

Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

San Francisco
Berlin
denial

mayor of arg1
live in arg1

Semi-Supervised Bootstrap Learning

<u>Learn which noun phrases are cities:</u>

Paris
Pittsburgh
Seattle
Montpelier

San Francisco
Berlin
denial

mayor of arg1
live in arg1

arg1 is home of
traits such as arg1

Semi-Supervised Bootstrap Learning

<u>Learn which noun phrases are cities:</u>

Paris
Pittsburgh
Seattle
Montpelier

mayor of arg1
live in arg1

San Francisco
Berlin
denial

arg1 is home of
traits such as arg1

anxiety
selfishness
London

Semi-Supervised Bootstrap Learning

it's underconstrained!!

Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

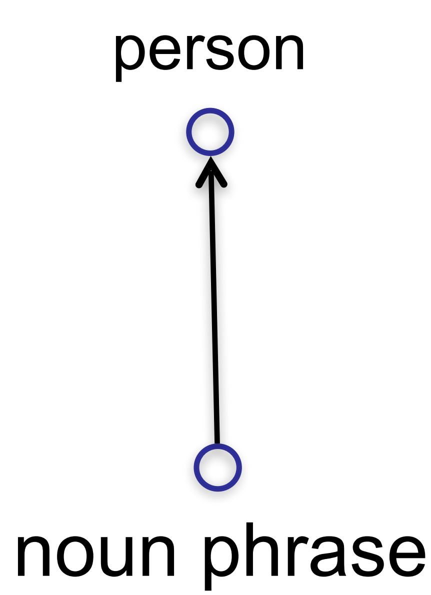
mayor of arg1
live in arg1

San Francisco
Berlin
denial

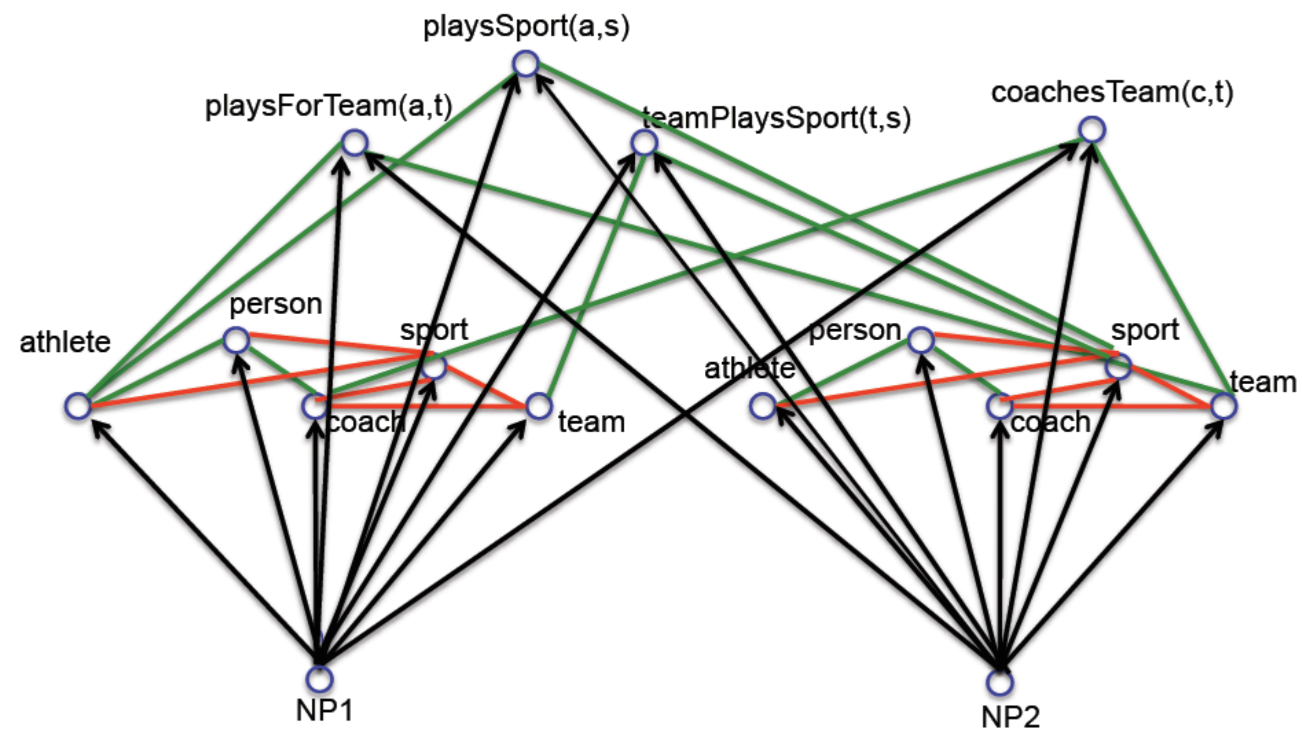
arg1 is home of
traits such as arg1

anxiety
selfishness
London

Key Idea 1: Coupled semi-supervised training of many functions



hard
(underconstrained)
semi-supervised
learning problem

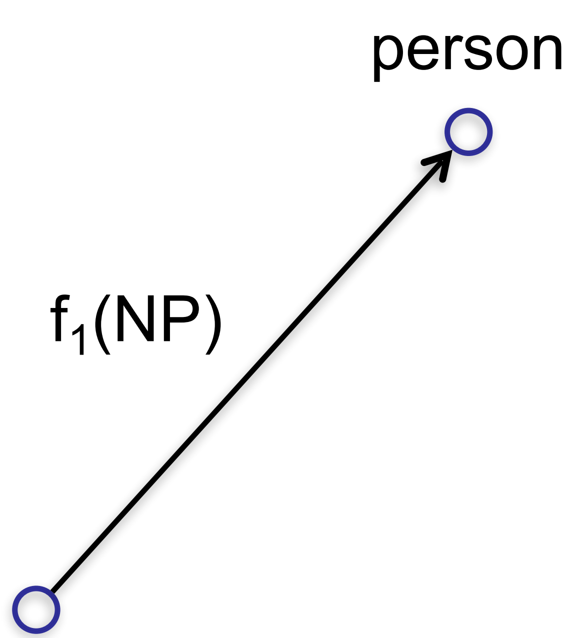


much easier (more constrained)
semi-supervised learning problem

Type 1 Coupling: Co-Training, Multi-View Learning

Supervised training of 1 function:

Minimize: $$\sum_{<np,person> \in \, labeled \ data} |f_1(np) - person|$$



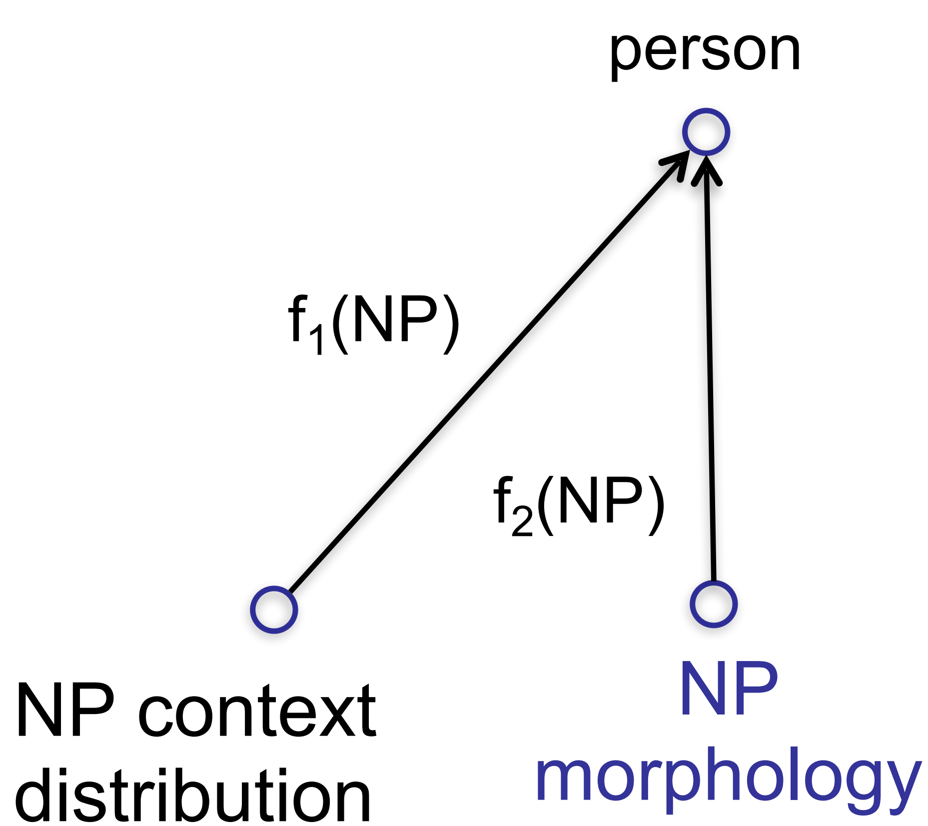
NP: NP context distribution

__ is a friend
rang the __
…
__ walked in

Type 1 Coupling: Co-Training, Multi-View Learning

Coupled training of 2 functions:

Minimize:
$$\sum_{<np,person>\ \in\ labeled\ data} |f_1(np) - person|$$
$$+ \sum_{<np,person>\ \in\ labeled\ data} |f_2(np) - person|$$
$$+ \sum_{np\ \in\ unlabeled\ data} |f_1(np) - f_2(np)|$$

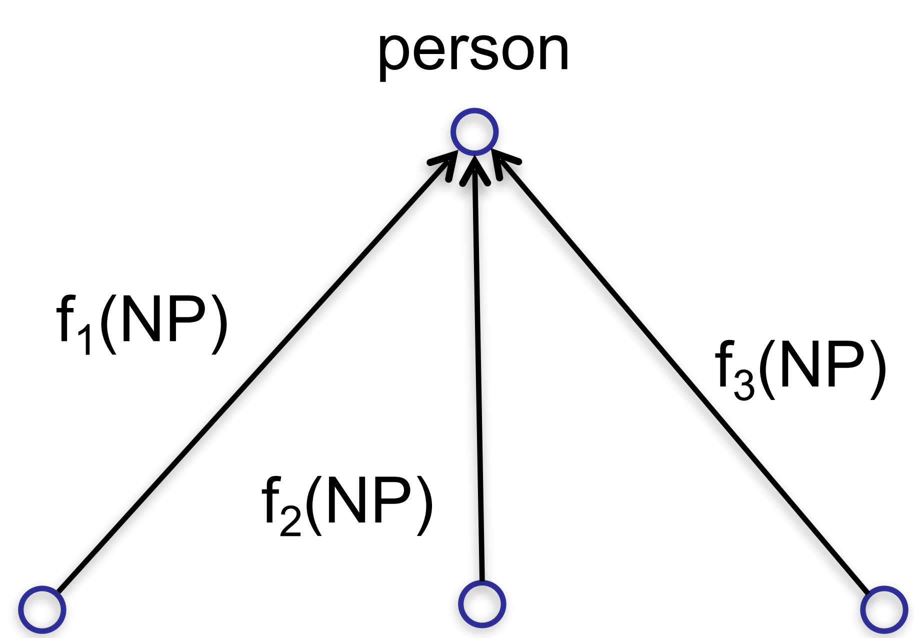


NP:

__ is a friend	***capitalized?***
rang the __	***ends with '...ski'?***
…	***…***
__ walked in	***contains "univ."?***

Type 1 Coupling: Co-Training, Multi-View Learning

[Blum & Mitchell; 98]
[Dasgupta et al; 01]
[Ganchev et al., 08]
[Sridharan & Kakade, 08]
[Wang & Zhou, ICML10]



NP: NP context distribution | NP morphology | NP HTML contexts

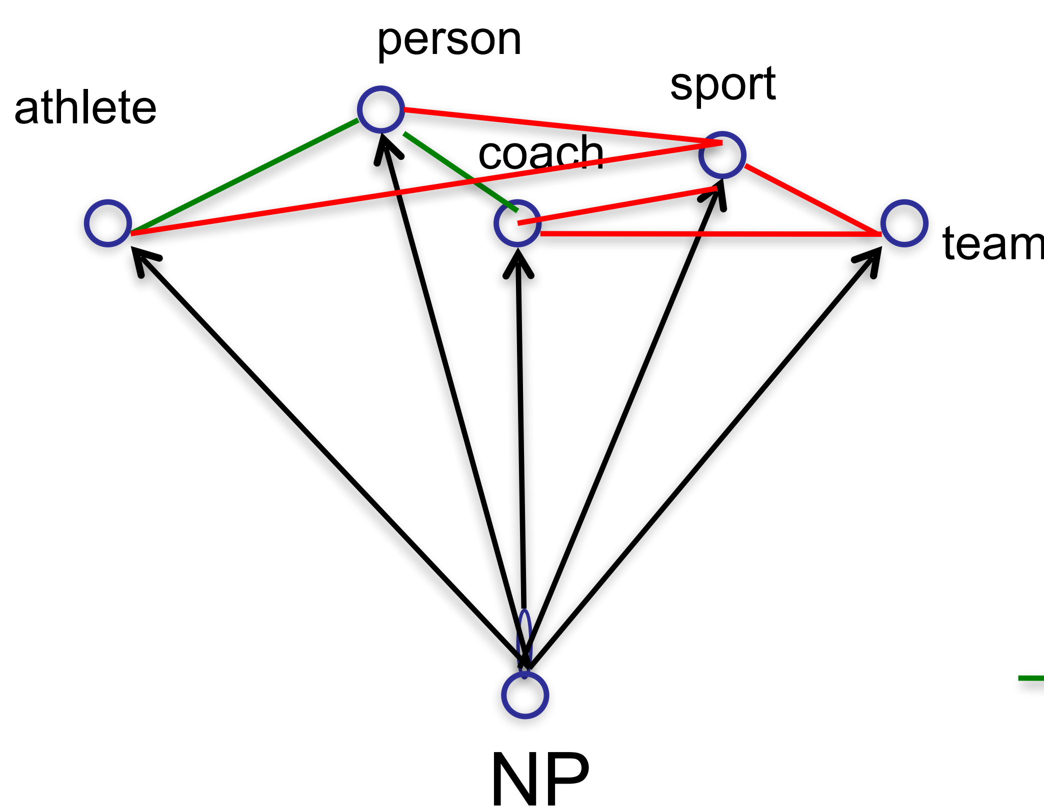
__ is a friend
rang the __
…
__ walked in

capitalized?
ends with '...ski'?
…
contains "univ."?

www.celebrities.com:
*** __ ***
…

Type 2 Coupling: Multi-task, Structured Outputs

[Daume, 2008]
[Bakhir et al., eds. 2007]
[Roth et al., 2008]
[Taskar et al., 2009]
[Carlson et al., 2009]

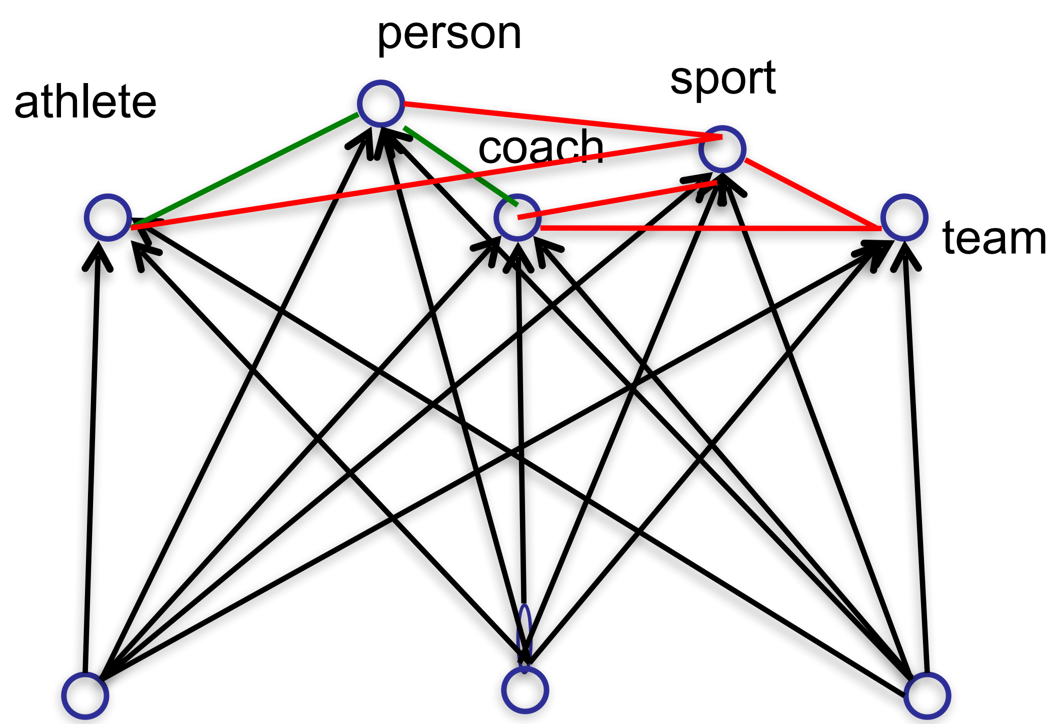


athlete(NP) → person(NP)

athlete(NP) → NOT sport(NP)

NOT athlete(NP) ← sport(NP)

Multi-view, Multi-Task Coupling



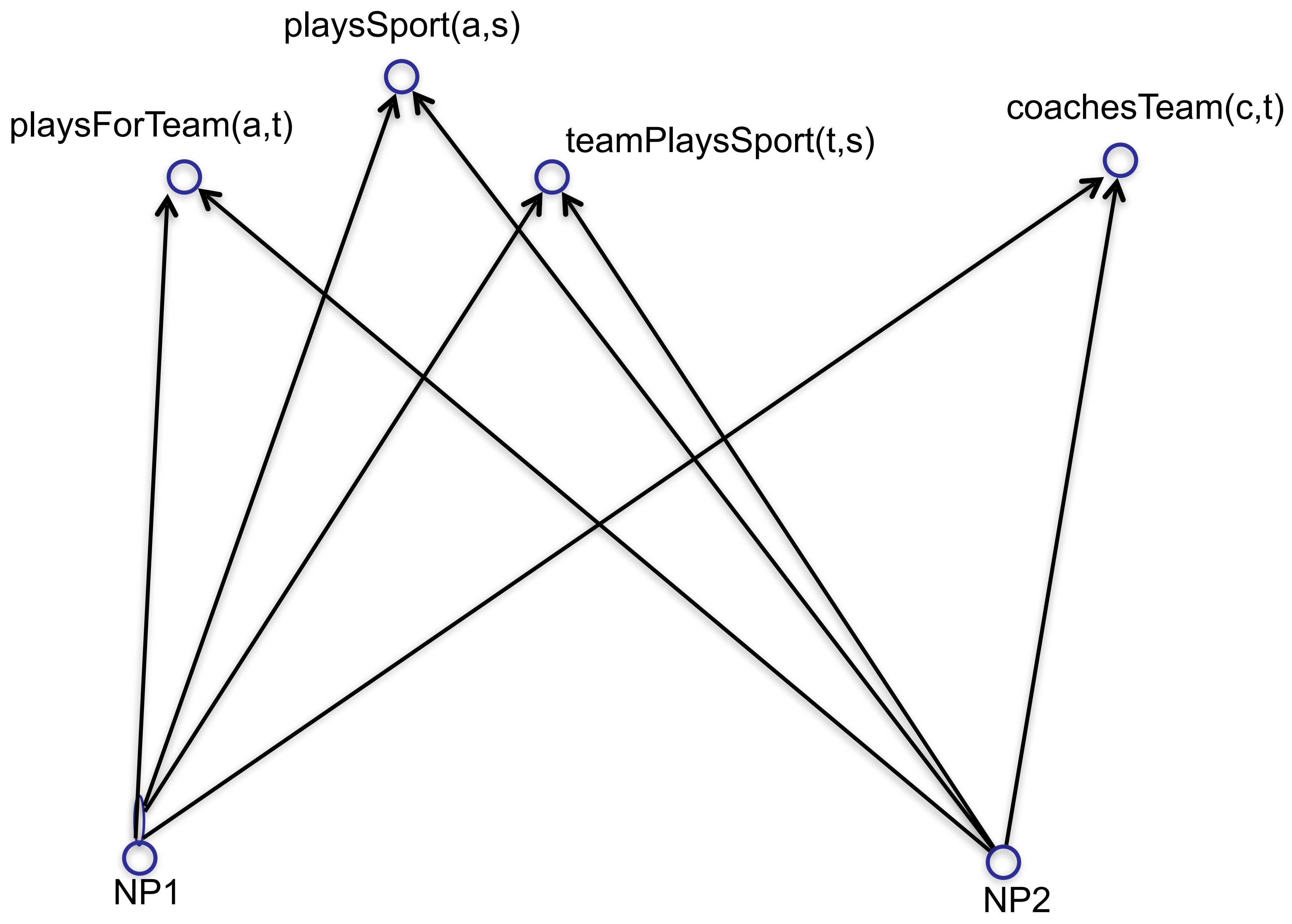
NP:

NP text context distribution

NP morphology

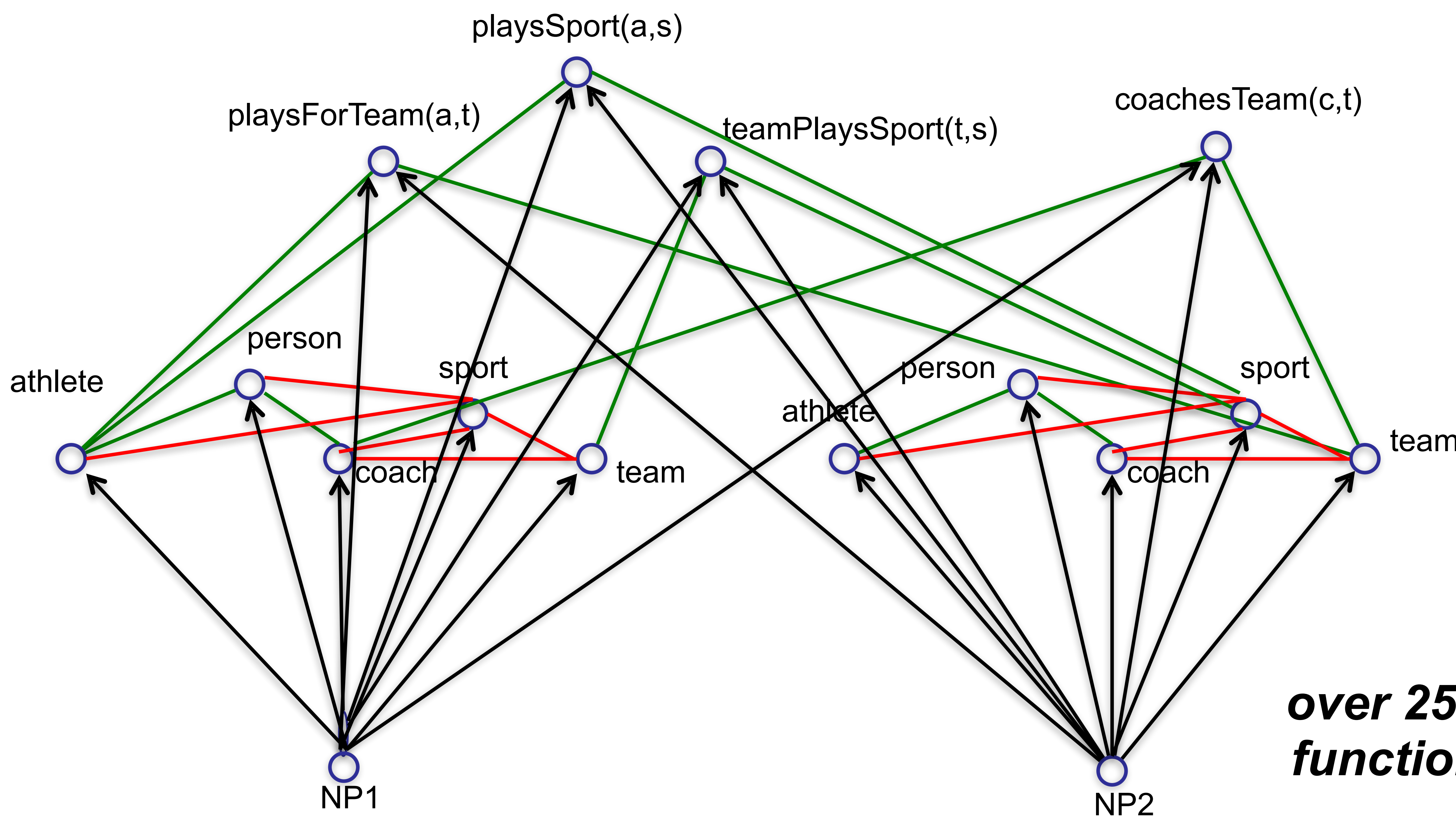
NP HTML contexts

Type 3 Coupling: Learning Relations



Type 3 Coupling: Argument Types

playsSport(NP1,NP2) → athlete(NP1), sport(NP2)



over 2500 coupled functions in NELL

NELL: Learned reading strategies

Plays_Sport(arg1,arg2):

arg1_was_playing_arg2   arg2_megastar_arg1   arg2_icons_arg1
arg2_player_named_arg1   arg2_prodigy_arg1
arg1_is_the_tiger_woods_of_arg2   arg2_career_of_arg1   arg2_greats_as_arg1
arg1_plays_arg2   arg2_player_is_arg1   arg2_legends_arg1
arg1_announced_his_retirement_from_arg2   arg2_operations_chief_arg1
arg2_player_like_arg1   arg2_and_golfing_personalities_including_arg1
arg2_players_like_arg1   arg2_greats_like_arg1
arg2_players_are_steffi_graf_and_arg1   arg2_great_arg1   arg2_champ_arg1
arg2_greats_such_as_arg1   arg2_professionals_such_as_arg1
arg2_hit_by_arg1 arg2_greats_arg1   arg2_icon_arg1   arg2_stars_like_arg1
arg2_pros_like_arg1   arg1_retires_from_arg2   arg2_phenom_arg1
arg2_lesson_from_arg1   arg2_architects_robert_trent_jones_and_arg1
arg2_sensation_arg1 arg2_pros_arg1   arg2_stars_venus_and_arg1
arg2_hall_of_famer_arg1 arg2_superstar_arg1   arg2_legend_arg1
arg2_legends_such_as_arg1   arg2_players_is_arg1   arg2_pro_arg1
arg2_player_was_arg1   arg2_god_arg1   arg2_idol_arg1
arg1_was_born_to_play_arg2   arg2_star_arg1   arg2_hero_arg1
arg2_players_are_arg1   arg1_retired_from_professional_arg2
arg2_legends_as_arg1   arg2_autographed_by_arg1 arg2_champion_arg1 …

NELL: Learned reading strategies

Plays_Sport(arg1,arg2):

arg1_was_playing_arg2 arg2_megasta
arg2_player_named_arg1 arg2_prodig
arg1_is_the_tiger_woods_of_arg2 arg2
arg1_plays_arg2 arg2_player_is_arg1
arg1_announced_his_retirement_from_a
arg2_player_like_arg1 arg2_and_golfin
arg2_players_like_arg1 arg2_greats_li
arg2_players_are_steffi_graf_and_arg1
arg2_greats_such_as_arg1 arg2_profe
arg2_hit_by_arg1 arg2_greats_arg1 ar
arg2_pros_like_arg1 arg1_retires_from
arg2_lesson_from_arg1 arg2_architect
arg2_sensation_arg1 arg2_pros_arg1 a
arg2_hall_of_famer_arg1 arg2_supersta
arg2_legends_such_as_arg1 arg2_play
arg2_player_was_arg1 arg2_god_arg1
arg1_was_born_to_play_arg2 arg2_sta
arg2_players_are_arg1 arg1_retired_fr
arg2_legends_as_arg1 arg2_autograp

Predicate	Feature	Weight
mountain	LAST=peak	1.791
mountain	LAST=mountain	1.093
mountain	FIRST=mountain	-0.875
musicArtist	LAST=band	1.853
musicArtist	POS=DT_NNS	1.412
musicArtist	POS=DT_JJ_NN	-0.807
newspaper	LAST=sun	1.330
newspaper	LAST=university	-0.318
newspaper	POS=NN_NNS	-0.798
university	LAST=college	2.076
university	PREFIX=uc	1.999
university	LAST=state	1.992
university	LAST=university	1.745
university	FIRST=college	-1.381
visualArtMovement	SUFFIX=ism	1.282
visualArtMovement	PREFIX=journ	-0.234
visualArtMovement	PREFIX=budd	-0.253

NELL: Learned reading strategies

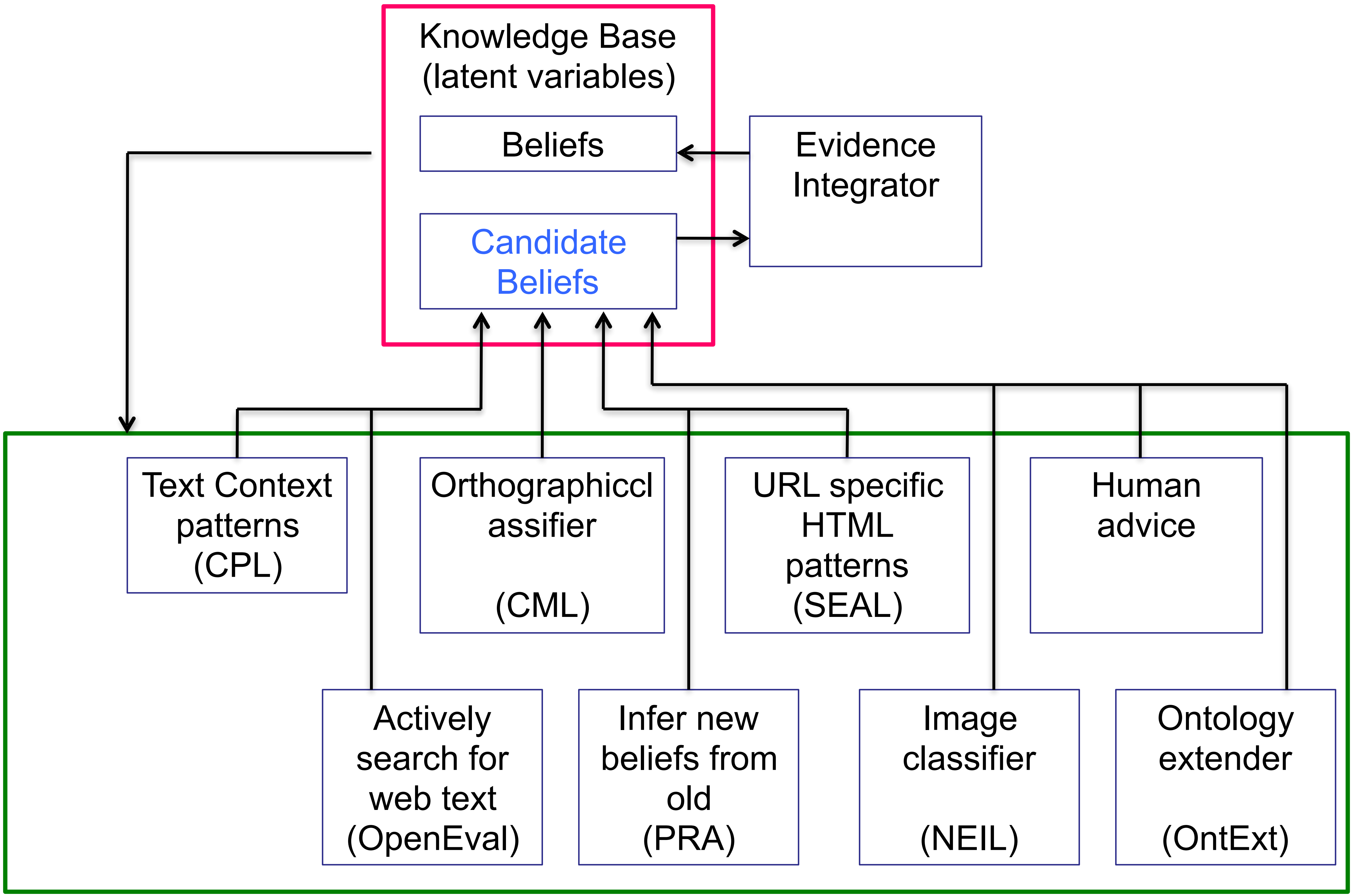
Plays_Sport(arg1,arg2):

arg1_was_playing_arg2 arg2_megasta
arg2_player_named_arg1 arg2_prodig
arg1_is_the_tiger_woods_of_arg2 arg2
arg1_plays_arg2 arg2_player_is_arg1
arg1_announced_his_retirement_from_a
arg2_player_like_arg1 arg2_and_golfi
arg2_players_like_arg1 arg2_greats_li
arg2_players_are_steffi_graf_and_arg1
arg2_greats_such_as_arg1 arg2_profe
arg2_hit_by_arg1 arg2_greats_arg1 ar
arg2_pros_like_arg1 arg1_retires_from
arg2_lesson_from_arg1 arg2_architect
arg2_sensation_arg1 arg2_pros_arg1 a
arg2_hall_of_famer_arg1 arg2_supersta
arg2_legends_such_as_arg1 arg2_pla
arg2_player_was_arg1 arg2_god_arg1
arg1_was_born_to_play_arg2 arg2_sta
arg2_players_are_arg1 arg1_retired_fr

Predicate	Feature	Weight
mountain	LAST=peak	1.791
mountain	LAST=mountain	1.093
mountain	FIRST=mountain	-0.875
musicArtist	LAST=band	1.853
musicArtist	POS=DT_NNS	1.412
musicArtist	POS=DT_JJ_NN	-0.807
newspaper	LAST=sun	1.330
newspaper	LAST=university	-0.318
newspaper	POS=NN_NNS	-0.798
university	LAST=college	2.076
university	PREFIX=uc	1.999
university	LAST=state	1.992
university	LAST=university	1.745
university	FIRST=college	-1.381
visualArtMovement	SUFFIX=ism	1.282

Predicate	Web URL	Extraction Template
academicField	http://scholendow.ais.msu.edu/student/ScholSearch.Asp	[X] -
athlete	http://www.quotes-search.com/d_occupation.aspx?o=+athlete	<a href='d_author.aspx?a=[X]'>-
bird	http://www.michaelforsberg.com/stock.html	<option>[X]</option>
bookAuthor	http://lifebehindthecurve.com/	[X] by [Y] –

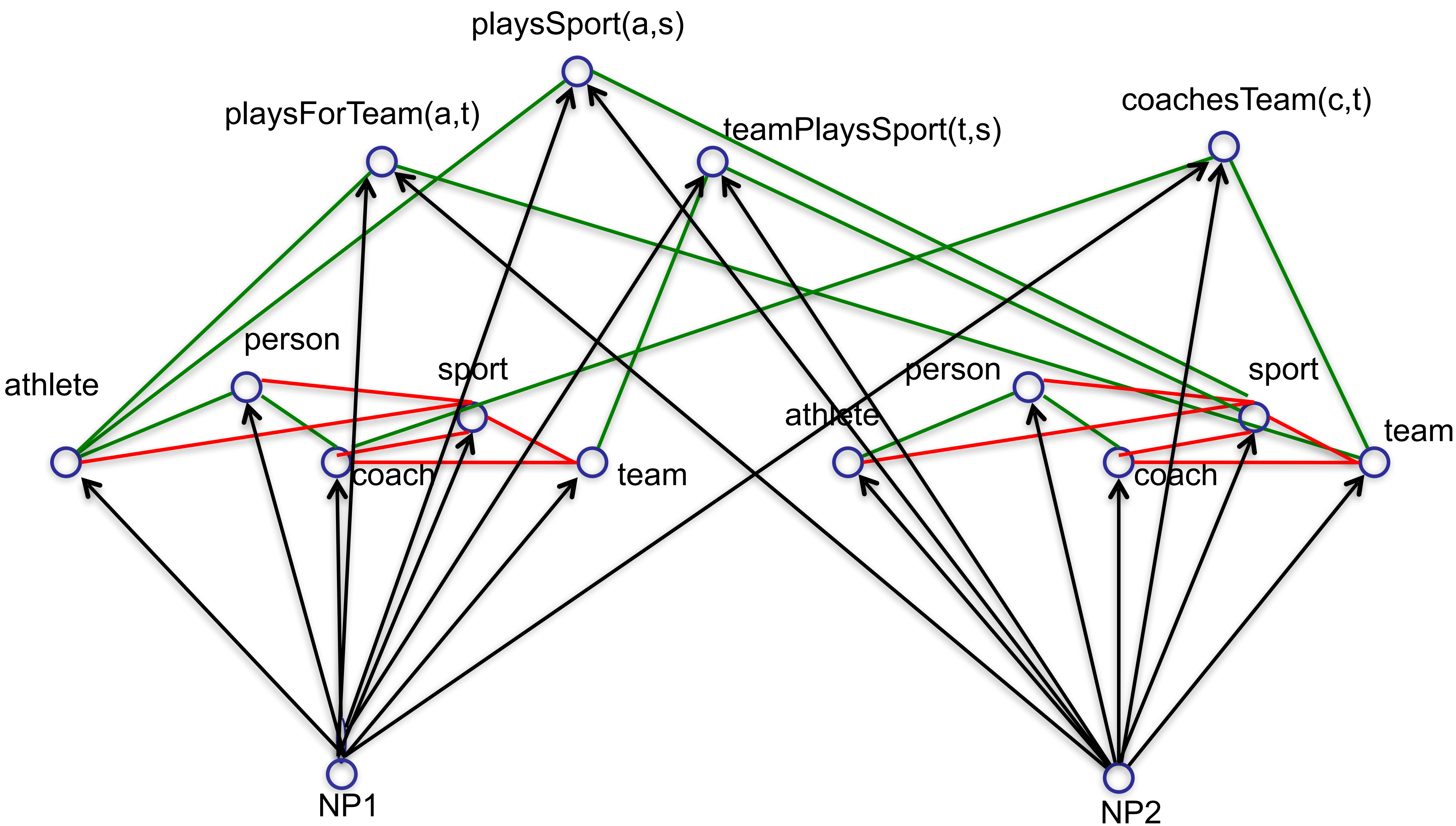
NELL Architecture



If coupled learning is the key,
how can we get new coupling constraints?

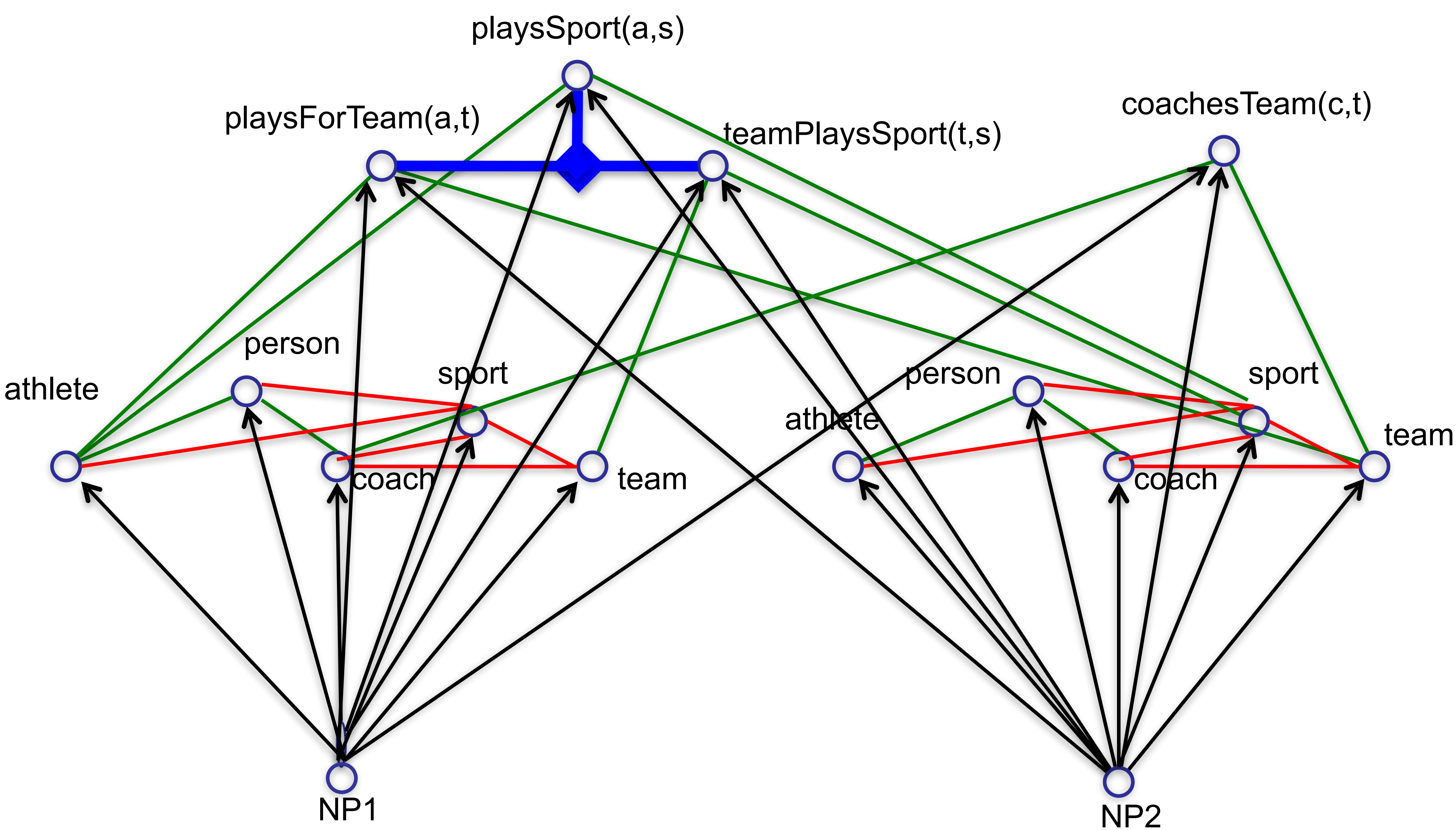
Learned Probabilistic Horn Clause Rules

0.93 playsSport(?x,?y) ← playsForTeam(?x,?z), teamPlaysSport(?z,?y)



Learned Probabilistic Horn Clause Rules

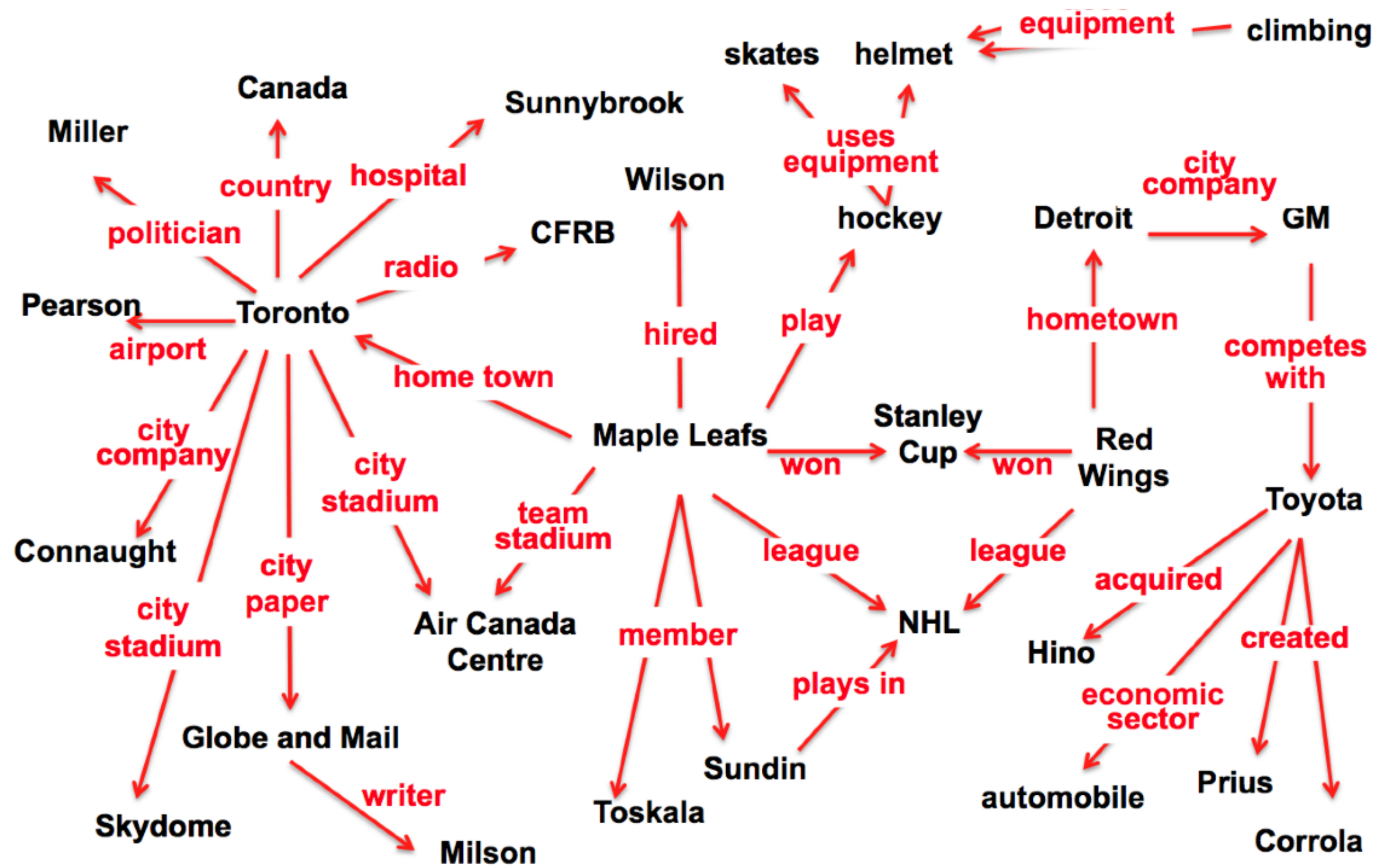
$\boxed{0.93 \text{ playsSport}(?x,?y) \leftarrow \text{playsForTeam}(?x,?z), \text{teamPlaysSport}(?z,?y)}$



Inference by KB Random Walks

[Lao et al, *EMNLP* 2011]

KB:



Random walk path type:

x — **competes with** → ? — **economic sector** → y

model Pr(R(x,y)): logistic function for R(x,y)

i^th feature: probability of arriving at node y starting at node x, and taking a random walk along path type i

CityLocatedInCountry(Pittsburgh) = ?

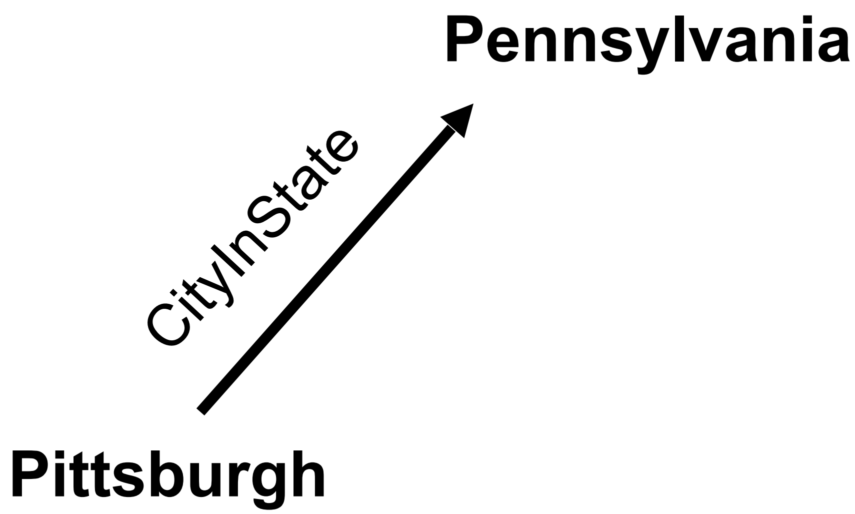
[Lao et al, *EMNLP* 2011]

Pittsburgh

Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32

CityLocatedInCountry(Pittsburgh) = ?

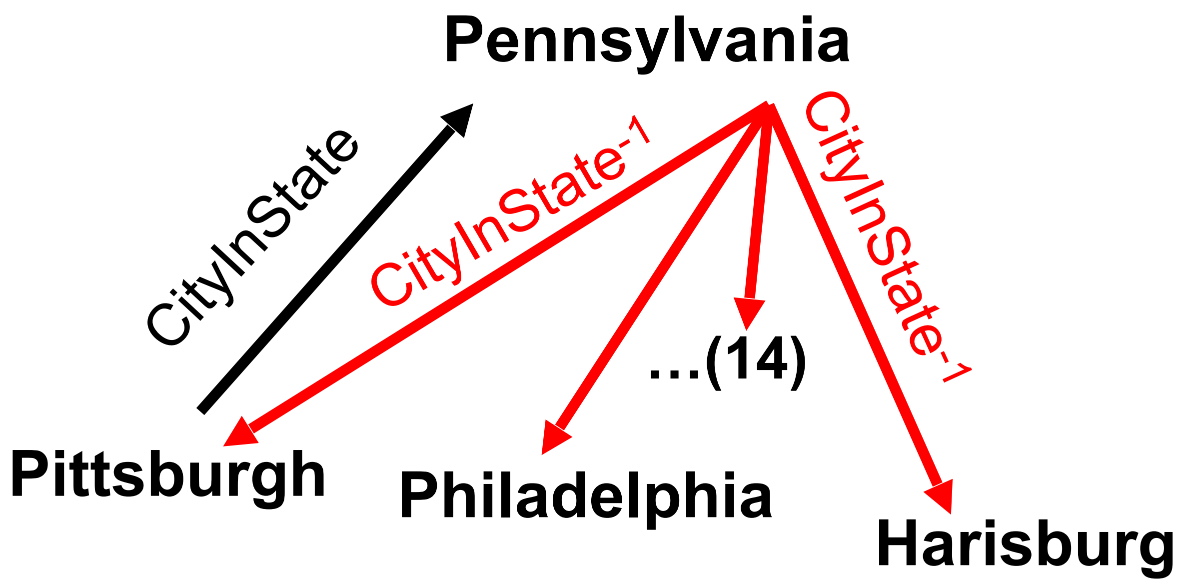
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32

CityLocatedInCountry(Pittsburgh) = ?

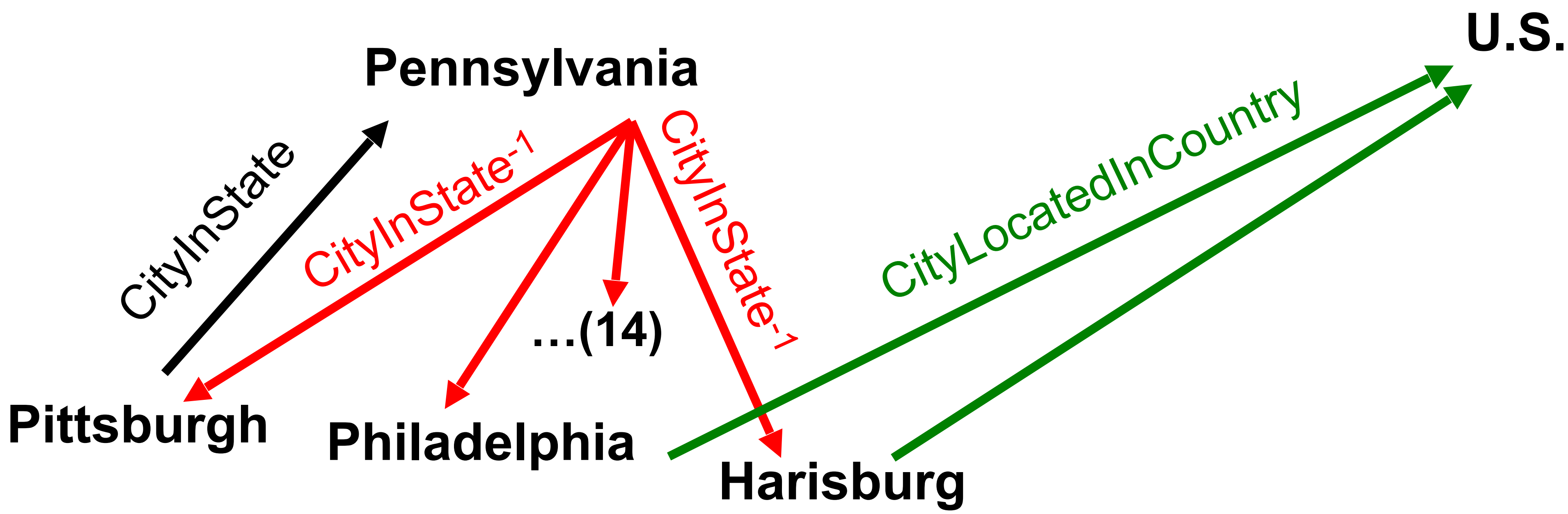
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32

CityLocatedInCountry(Pittsburgh) = ?

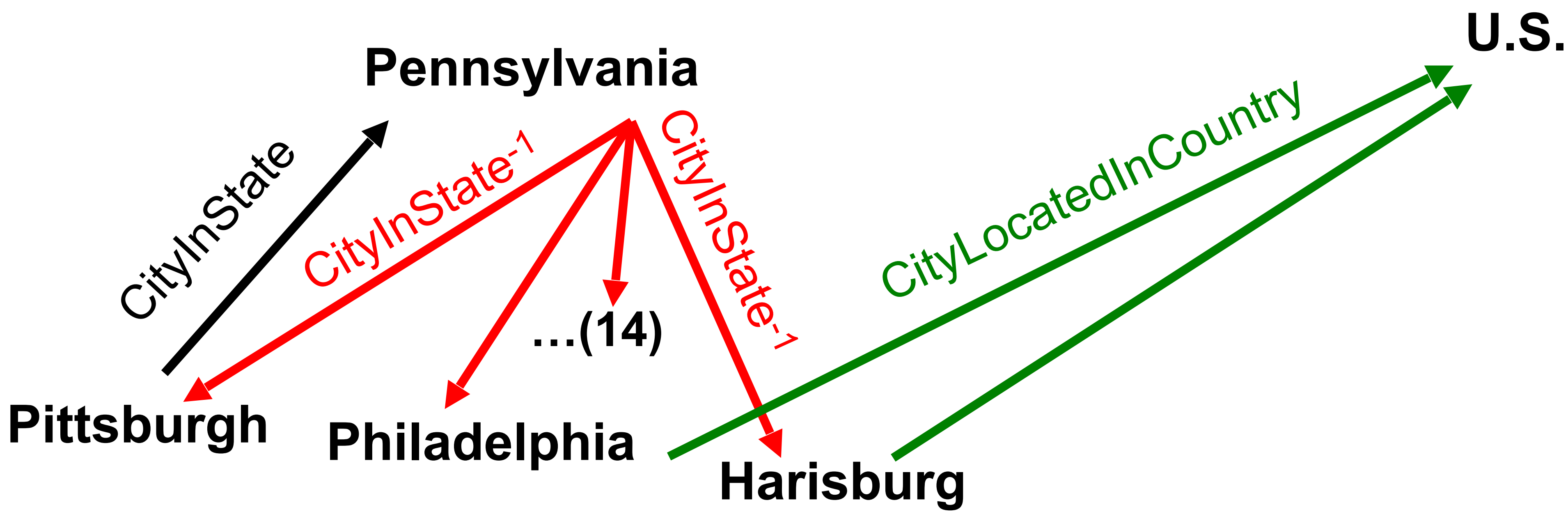
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	Feature Value	Logistic Regresssion Weight
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32

CityLocatedInCountry(Pittsburgh) = ?

[Lao et al, *EMNLP* 2011]

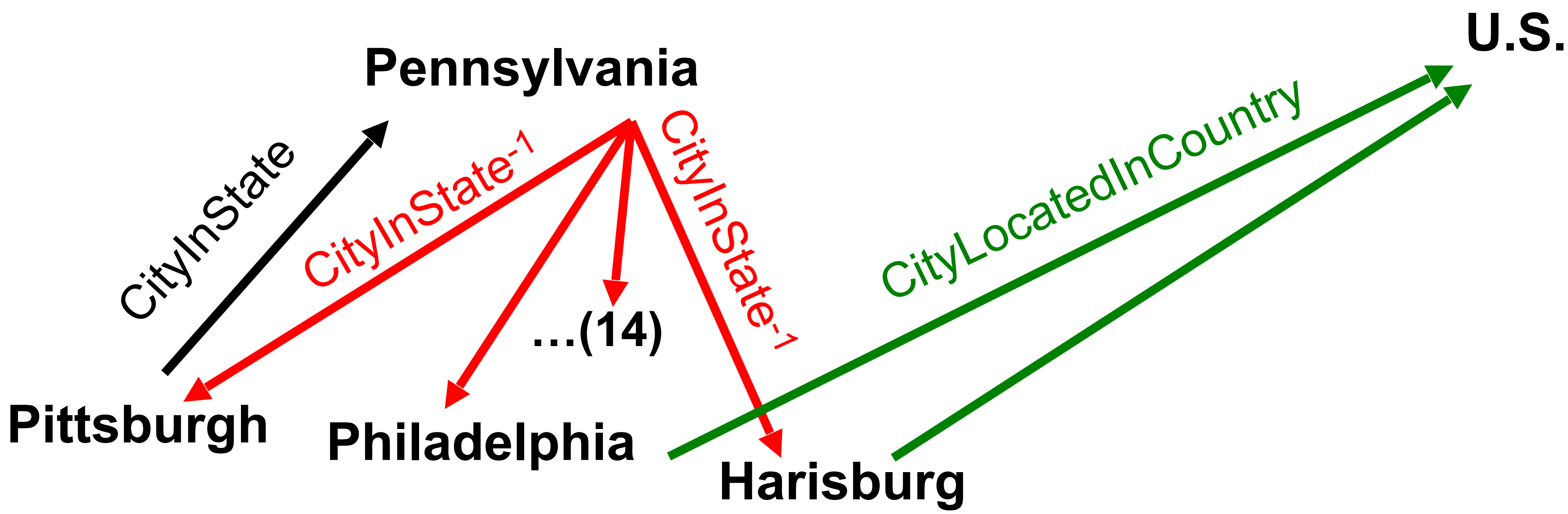


Pr(U.S. | Pittsburgh, TypedPath)

Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32

CityLocatedInCountry(Pittsburgh) = ?

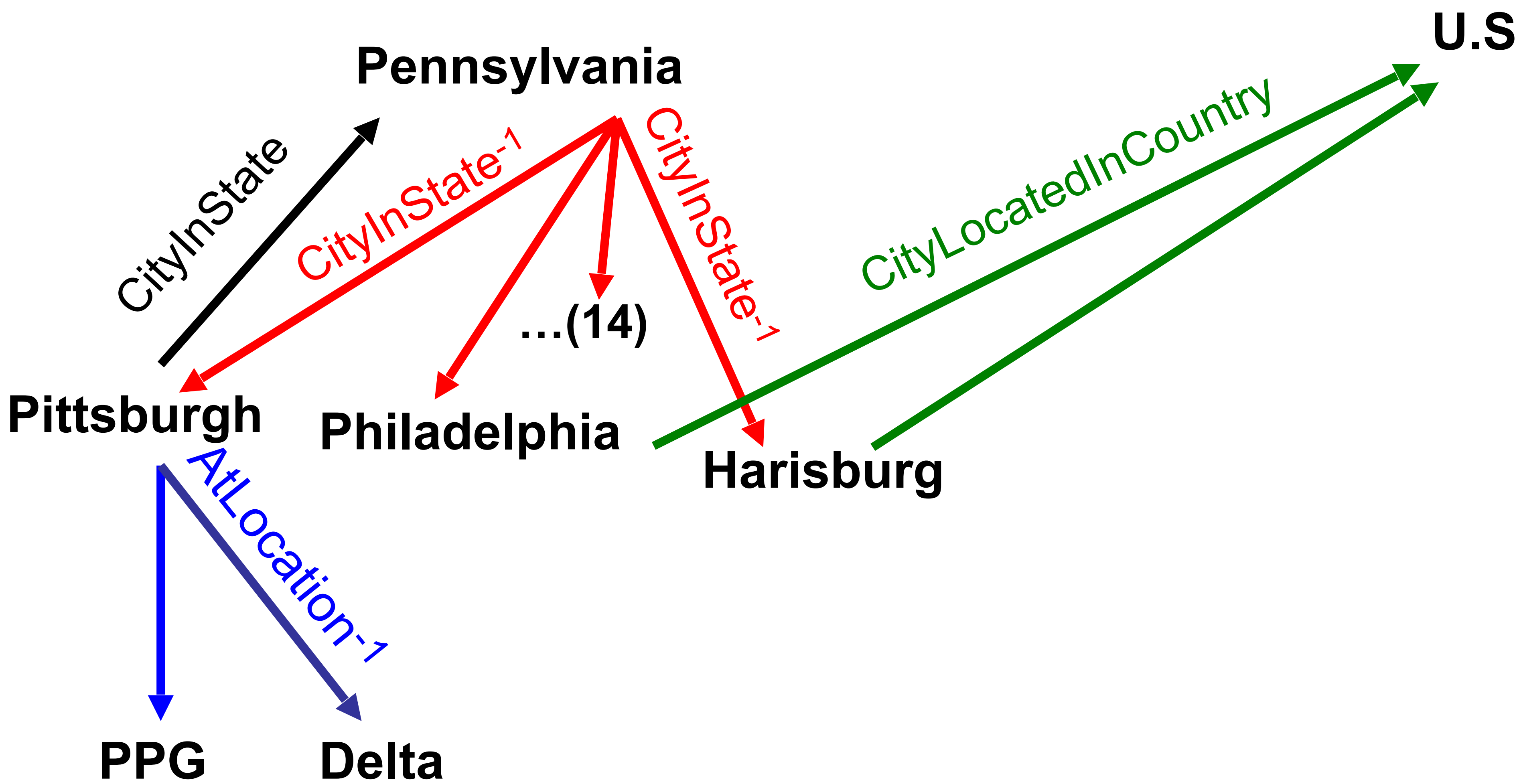
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	Feature Value	Logistic Regresssion Weight
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry		0.20

CityLocatedInCountry(Pittsburgh) = ?

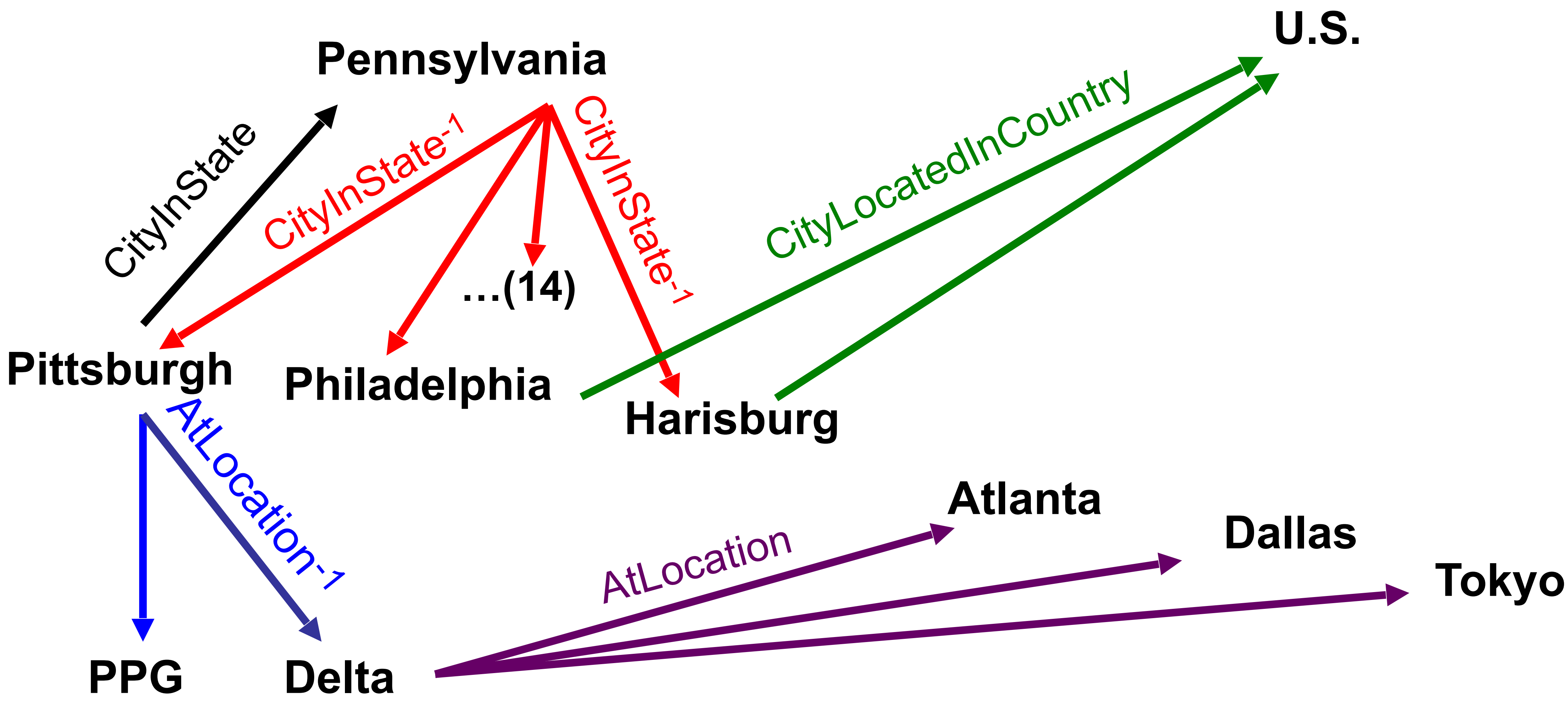
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry		0.20

CityLocatedInCountry(Pittsburgh) = ?

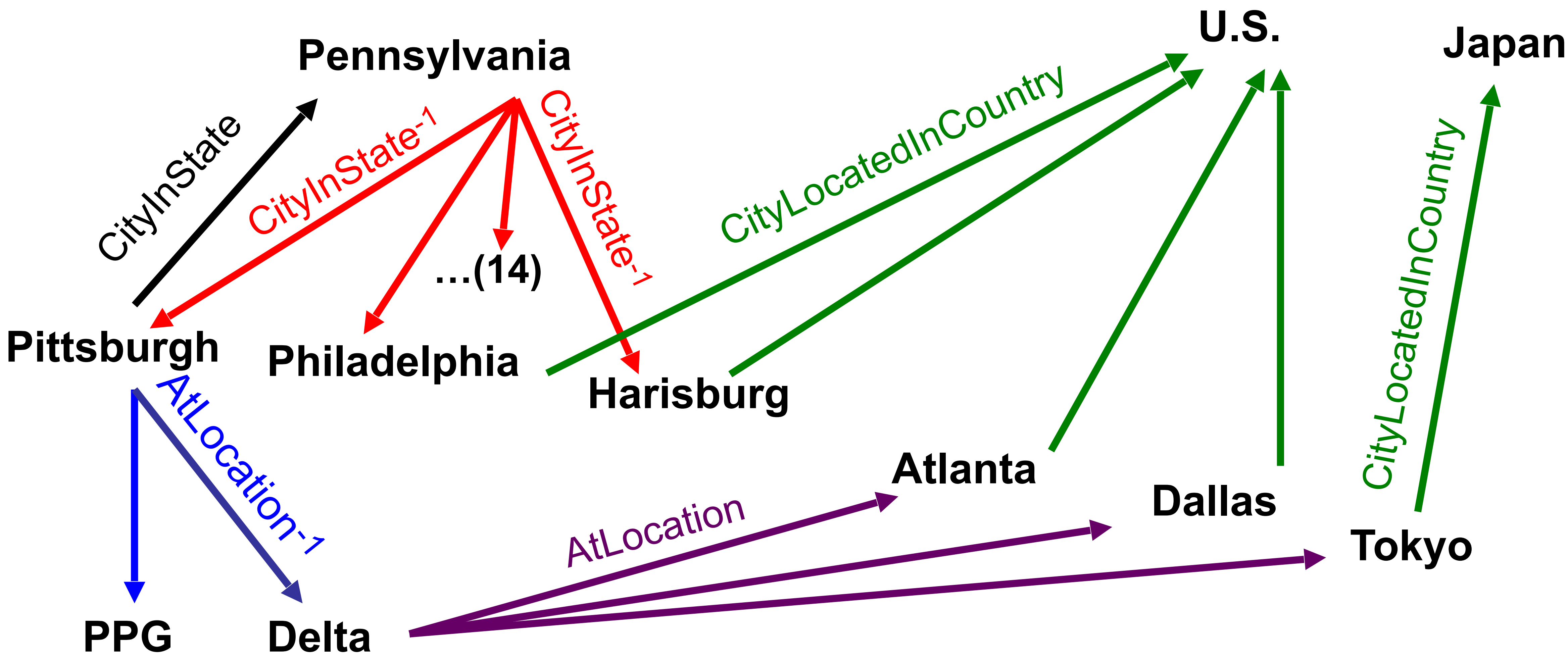
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry		0.20

CityLocatedInCountry(Pittsburgh) = ?

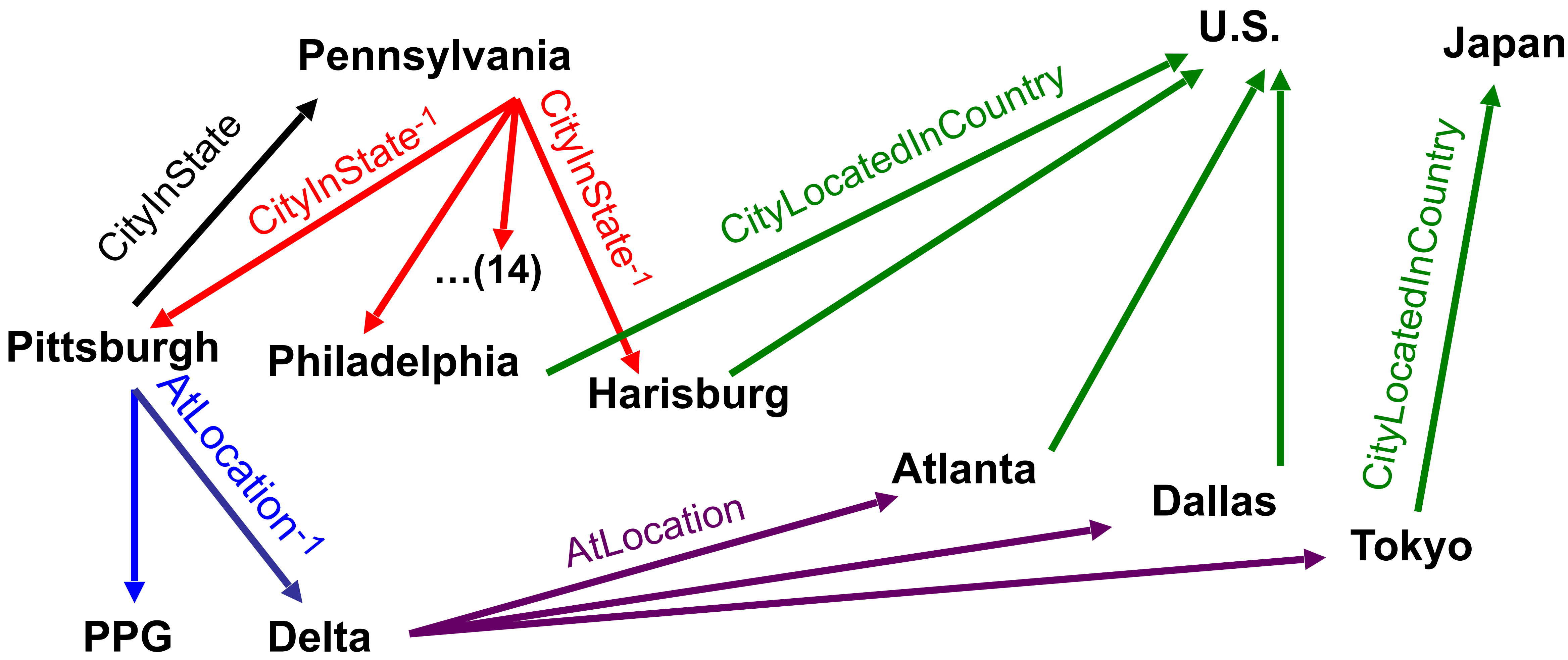
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	Feature Value	Logistic Regresssion Weight
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry	0.6	0.20

CityLocatedInCountry(Pittsburgh) = ?

[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry	0.6	0.20
…	…	…

CityLocatedInCountry(Pittsburgh) = U.S.   p=0.58

Random walk inference: learned path types

CityLocatedInCountry(*city, country*):

8.04 cityliesonriver, cityliesonriver^{-1}, citylocatedincountry

5.42 hasofficeincity^{-1}, hasofficeincity, citylocatedincountry

4.98 cityalsoknownas, cityalsoknownas, citylocatedincountry

2.85 citycapitalofcountry,citylocatedincountry^{-1},citylocatedincountry

2.29 agentactsinlocation^{-1}, agentactsinlocation, citylocatedincountry

1.22 statehascapital^{-1}, statelocatedincountry

0.66 citycapitalofcountry

⋮

7 of the 2985 learned paths for CityLocatedInCountry

Key Idea 3:

Automatically extend ontology

Example Discovered Relations

[Mohamed et al. *EMNLP* 2011]

Category Pair	Frequent Instance Pairs	Text Contexts	Suggested Name
MusicInstrument Musician	sitar, George Harrison tenor sax, Stan Getz trombone, Tommy Dorsey vibes, Lionel Hampton	ARG1 master ARG2 ARG1 virtuoso ARG2 ARG1 legend ARG2 ARG2 plays ARG1	Master
Disease Disease	pinched nerve, herniated disk tennis elbow, tendonitis blepharospasm, dystonia	ARG1 is due to ARG2 ARG1 is caused by ARG2	IsDueTo
CellType Chemical	epithelial cells, surfactant neurons, serotonin mast cells, histomine	ARG1 that release ARG2 ARG2 releasing ARG1	ThatReleas
Mammals Plant	koala bears, eucalyptus sheep, grasses goats, saplings	ARG1 eat ARG2 ARG2 eating ARG1	Eat
River City	Seine, Paris Nile, Cairo Tiber river, Rome	ARG1 in heart of ARG2 ARG1 which flows through ARG2	InHeartOf

NELL: sample of self-added relations

- athleteWonAward
- animalEatsFood
- languageTaughtInCity
- clothingMadeFromPlant
- beverageServedWithFood
- fishServedWithFood
- athleteBeatAthlete
- athleteInjuredBodyPart
- arthropodFeedsOnInsect
- animalEatsVegetable
- plantRepresentsEmotion
- foodDecreasesRiskOfDisease
- clothingGoesWithClothing
- bacteriaCausesPhysCondition
- buildingMadeOfMaterial
- emotionAssociatedWithDisease
- foodCanCauseDisease
- agriculturalProductAttractsInsect
- arteryArisesFromArtery
- countryHasSportsFans
- bakedGoodServedWithBeverage
- beverageContainsProtein
- animalCanDevelopDisease
- beverageMadeFromBeverage

Key Idea 4: Cumulative, Staged Learning

Learning X improves ability to learn Y

1. Classify noun phrases (NP's) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances
4. Learn which NP's (co)refer to which latent concepts
5. Discover new relations to extend ontology
6. Learn to infer relation instances via targeted random walks
7. Learn to assign temporal scope to beliefs
8. Learn to microread single sentences
9. Vision: co-train text and visual object recognition
10. Goal-driven reading: predict, then read to corroborate/correct
11. Make NELL a conversational agent on Twitter
12. Add a robot body to NELL

NELL Summary

- Learning
  - Coupled multi-task, multi-view semi-supervised training
- Inference
  - Data mine the KB to learn inference rules
  - Scalable any-time inference via random walks
- Representation
  - Ontology extension
    - invent new categories and relations
    - combine statistical clustering with direct reading
  - Infer millions of latent concepts from observable text
- Curriculum
  - learn easiest things first, build on those to “learn to learn”

Outline

13:00-13:15 Overview and motivation

13:15-13:45 Case study: NELL

13:45-14:00 Bootstrapped Entity Extraction

14:00-15:00 Open Relation Extraction & Canonicalization

15:00-15:30 **Coffee Break**

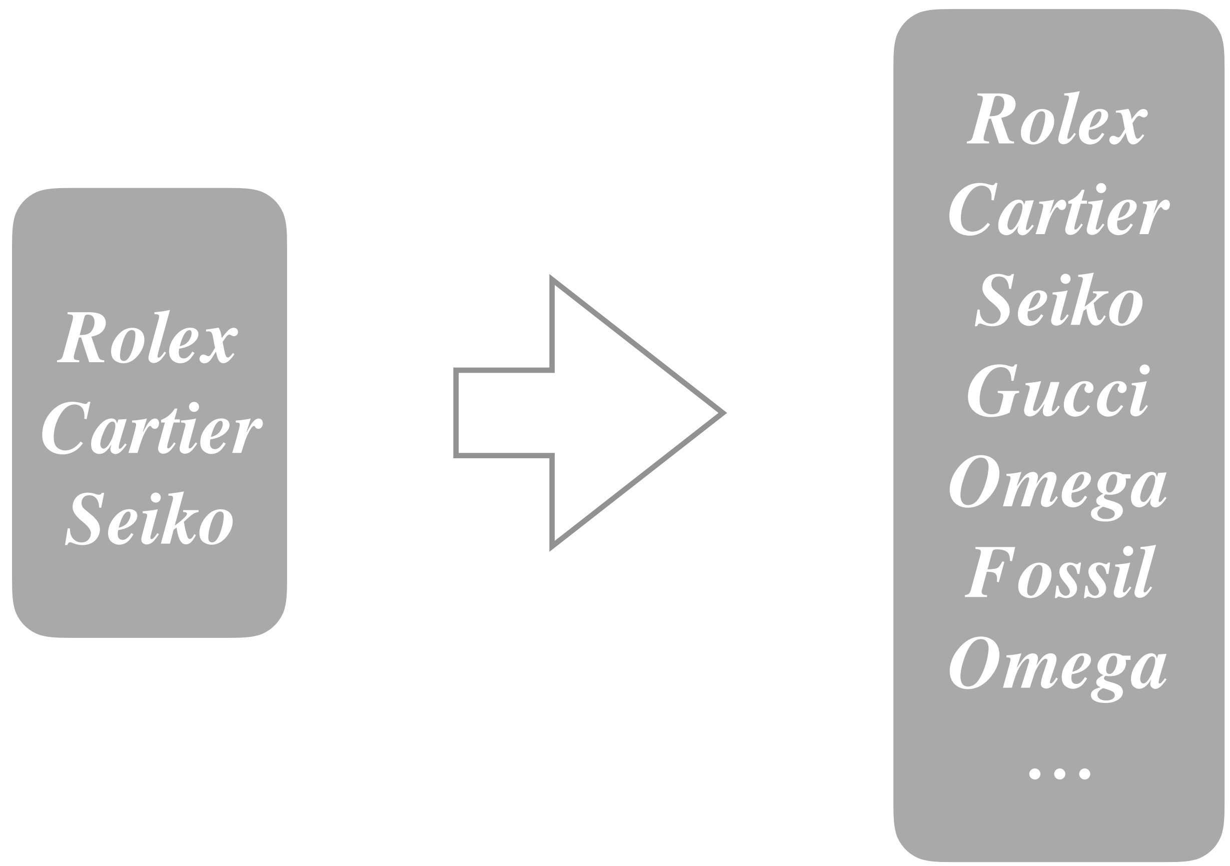
15:30-16:15 Distantly-supervised Relation Extraction

16:15-16:45 Knowledge Graph Embeddings

16:45-17:00 Conclusion & QA

Set Expansion

Given seed instances from a class, automatically identify more instances from that class

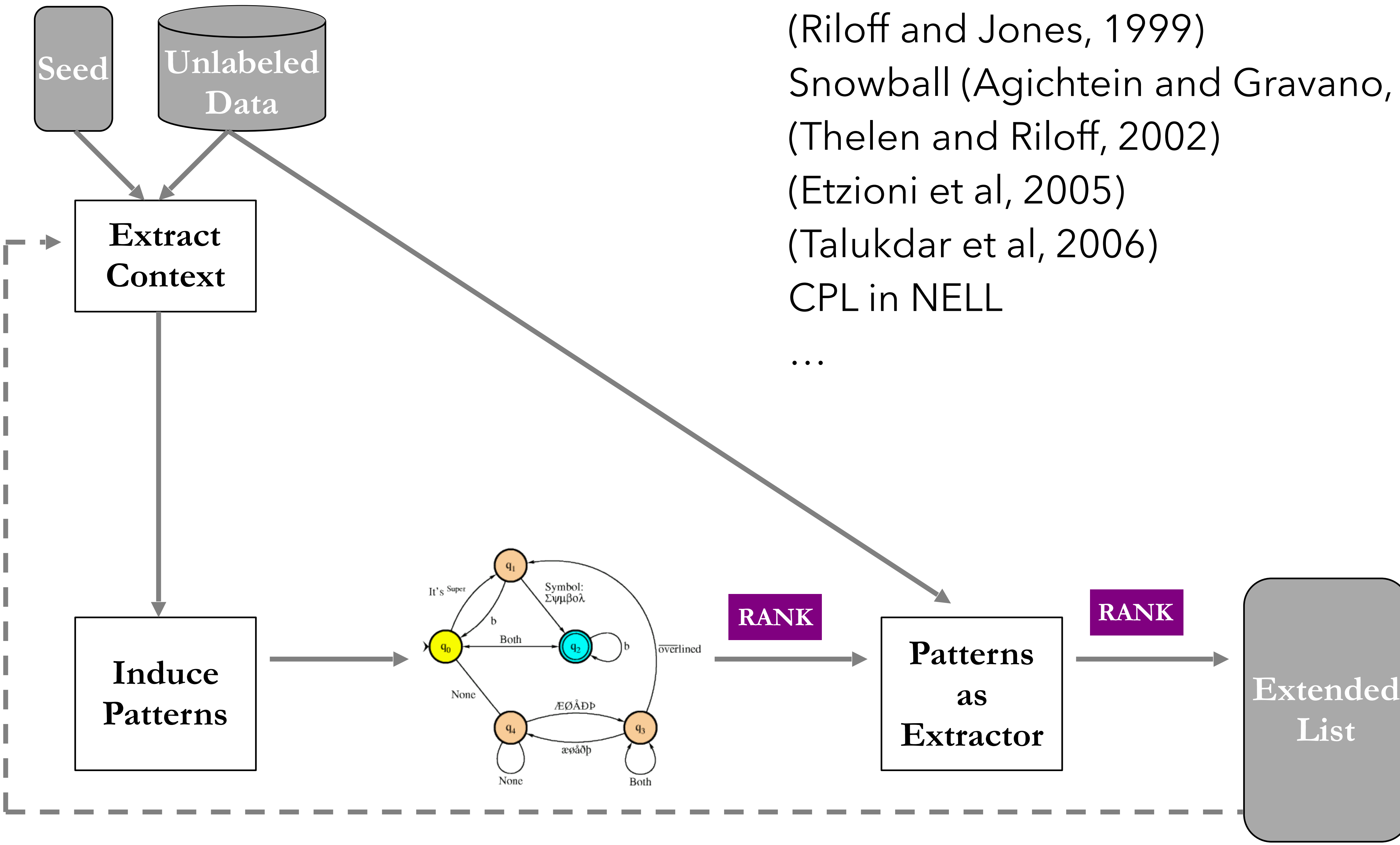


Many applications:

web advertising, knowledge graph population, …

Context Pattern Induction

DIPRE (Brin)
(Riloff and Jones, 1999)
Snowball (Agichtein and Gravano, 2000)
(Thelen and Riloff, 2002)
(Etzioni et al, 2005)
(Talukdar et al, 2006)
CPL in NELL
…



Extractions using Context Patterns

Induced Patterns (containing sequence "*watch*")

gold -<ENT>- watch
diamond -<ENT>- watch
fake -<ENT>- watches
bought -<ENT>- watch
encrusted -<ENT>- watch
stole -<ENT>- watch
Richemont AG , -<ENT>- watches
Rolex and -<ENT>- watches
buy -<ENT>- watches
Cartier and -<ENT>- watches
buy -<ENT>- watch
gold -<ENT>- watches

Richemont , -<ENT>- watches
bought -<ENT>- watches
fake -<ENT>- watch
diamond -<ENT>- watches
stole -<ENT>- watches
buy a -<ENT>- watch
jewelry , including -<ENT>- watch
watchmaker -<ENT>- .
jewelry , including -<ENT>- watches
stole a -<ENT>- watch
Rolex watches and -<ENT>- .
watchmaker -<ENT>- Group

Rolex watches are sold through official -<ENT>- and
bought a -<ENT>- watch
watchmaker -<ENT>- SA
Ulysse -<ENT>- watches
Rolex watches and -<ENT>- watch
Rolex , -<ENT>- watch
Rolex and -<ENT>- watch
diamond - studded -<ENT>- watch
diamond - encrusted -<ENT>- watch
Cartier , and -<ENT>- watches
buy a -<ENT>- watches
bought a -<ENT>- watches

Extracted Lists Improve NER Taggers

Training Data (Tokens)	Test-a		
	No List	Seed List	Unsup. List
9229	68.27	70.93	**72.26**
204657	89.52	84.30	**90.48**

Extractions using Context Patterns

Induced Patterns (containing sequence "*watch*")

gold -<*ENT*>- watch	Richemont , -<*ENT*>- watches	Rolex watches are sold through official -<*ENT*>- and
diamond -<*ENT*>- watch	bought -<*ENT*>- watches	bought a -<*ENT*>- watch
fake -<*ENT*>- watches	fake -<*ENT*>- watch	watchmaker -<*ENT*>- SA
bought -<*ENT*>- watch	diamond -<*ENT*>- watches	Ulysse -<*ENT*>- watches
encrusted -<*ENT*>- watch	stole -<*ENT*>- watches	Rolex watches and -<*ENT*>- watch
stole -<*ENT*>- watch	buy a -<*ENT*>- watch	Rolex , -<*ENT*>- watch
Richemont AG , -<*ENT*>- watches	jewelry , including -<*ENT*>- watch	Rolex and -<*ENT*>- watch
Rolex and -<*ENT*>- watches	watchmaker -<*ENT*>- .	diamond - studded -<*ENT*>- watch
buy -<*ENT*>- watches	jewelry , including -<*ENT*>- watches	diamond - encrusted -<*ENT*>- watch
Cartier and -<*ENT*>- watches	stole a -<*ENT*>- watch	Cartier , and -<*ENT*>- watches
buy -<*ENT*>- watch	Rolex watches and -<*ENT*>- .	buy a -<*ENT*>- watches
gold -<*ENT*>- watches	watchmaker -<*ENT*>- Group	bought a -<*ENT*>- watches

Entities Extracted by Above Patterns (ranked)

Rolex *(most confident)*	Fossil	Swatch
Cartier	Tag Heuer	Super Bowl
Swiss	Chanel	SPOT
Movado	Tiffany	Sekonda
Seiko	TechnoMarine	Rolexes
Gucci	Franck Muller	Harry Winston
Patek Philippe	Versace	Hampton Spirit
Piaget	Raymond Weil	Girard Perregaux
Omega	Guess	Frank Mueller
Citizen	Croton	David Yurman
Armani	Audemars Piguet	Chopard
DVD	DVDs	Chinese
Breitling	Montres Rolex	Armitron
Tourneau	CD	NFL *(least confident)*

Extracted Lists Improve NER Taggers

Training Data (Tokens)	Test-a		
	No List	Seed List	Unsup. List
9229	68.27	70.93	**72.26**
204657	89.52	84.30	**90.48**

SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

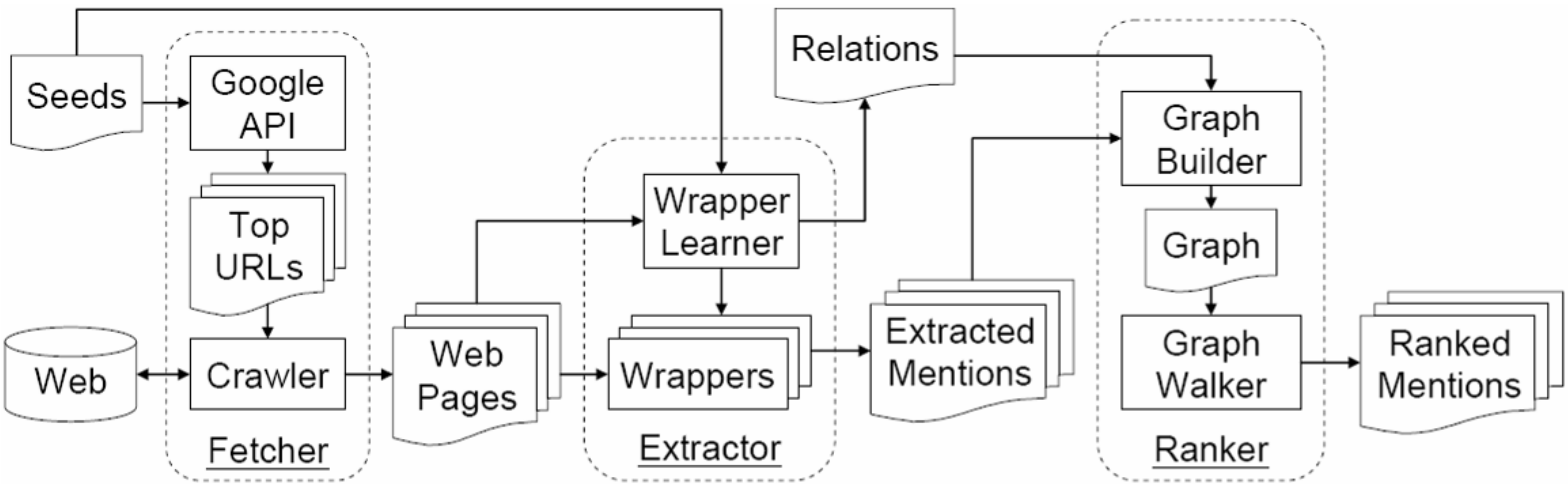


Figure 1. Flow chart of the SEAL system

- **Fetcher**: download web pages from the Web
- **Extractor**: learn wrappers from web pages
- **Ranker**: rank entities extracted by wrappers

SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

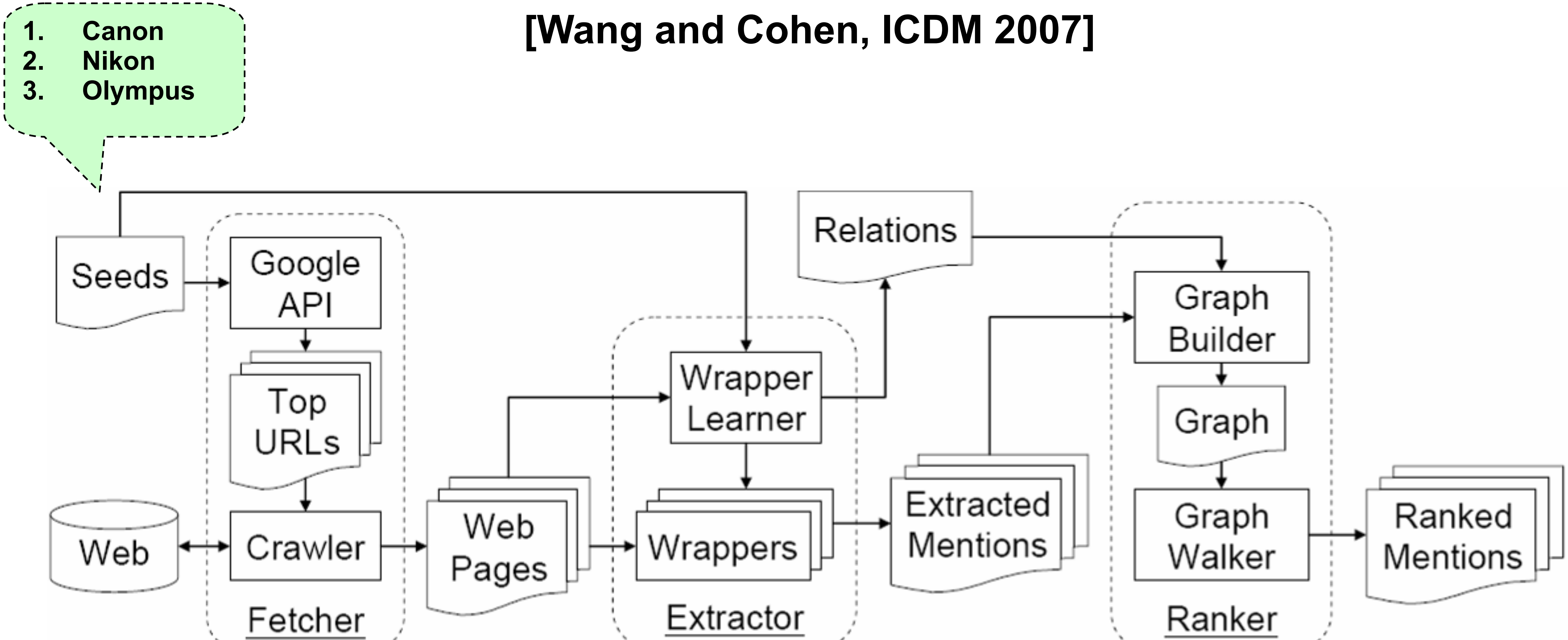


Figure 1. Flow chart of the SEAL system

- **Fetcher**: download web pages from the Web
- **Extractor**: learn wrappers from web pages
- **Ranker**: rank entities extracted by wrappers

SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

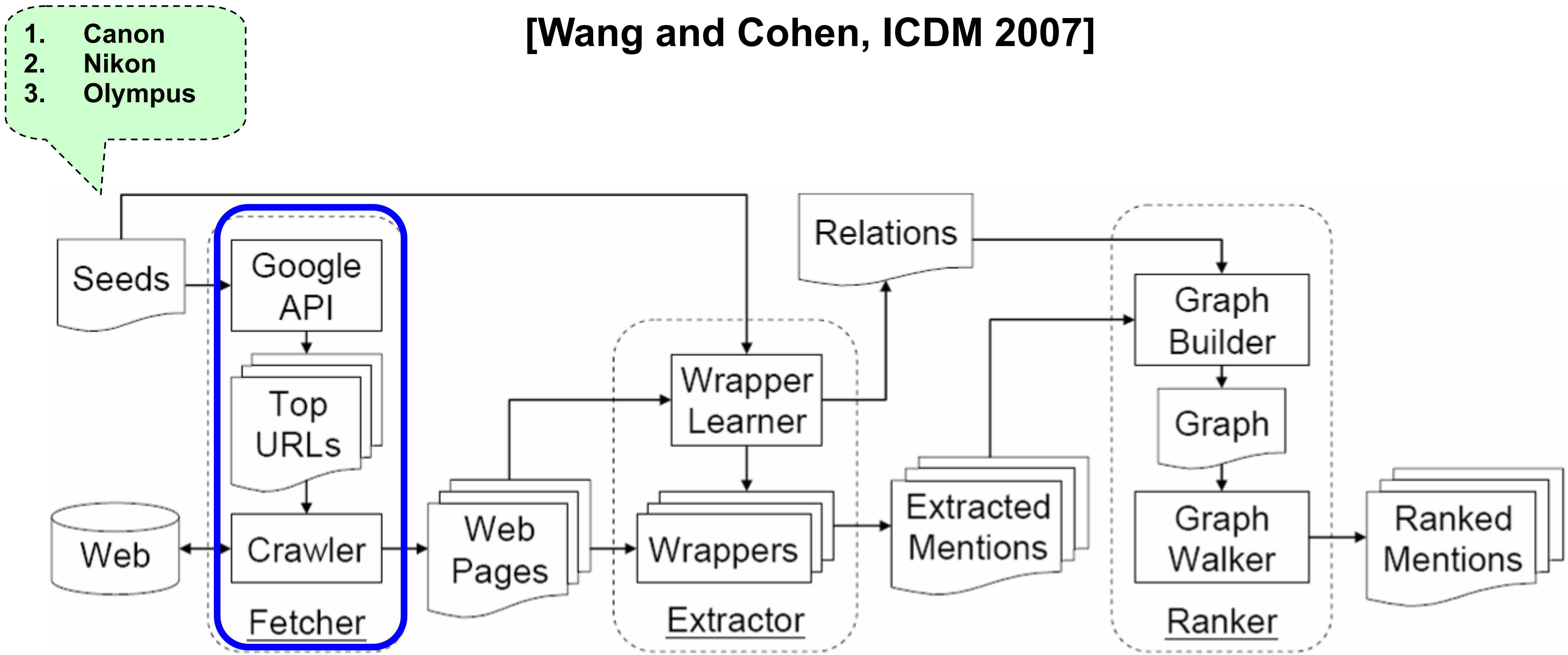


Figure 1. Flow chart of the SEAL system

- **Fetcher**: download web pages from the Web
- **Extractor**: learn wrappers from web pages
- **Ranker**: rank entities extracted by wrappers

SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

1. Canon
2. Nikon
3. Olympus

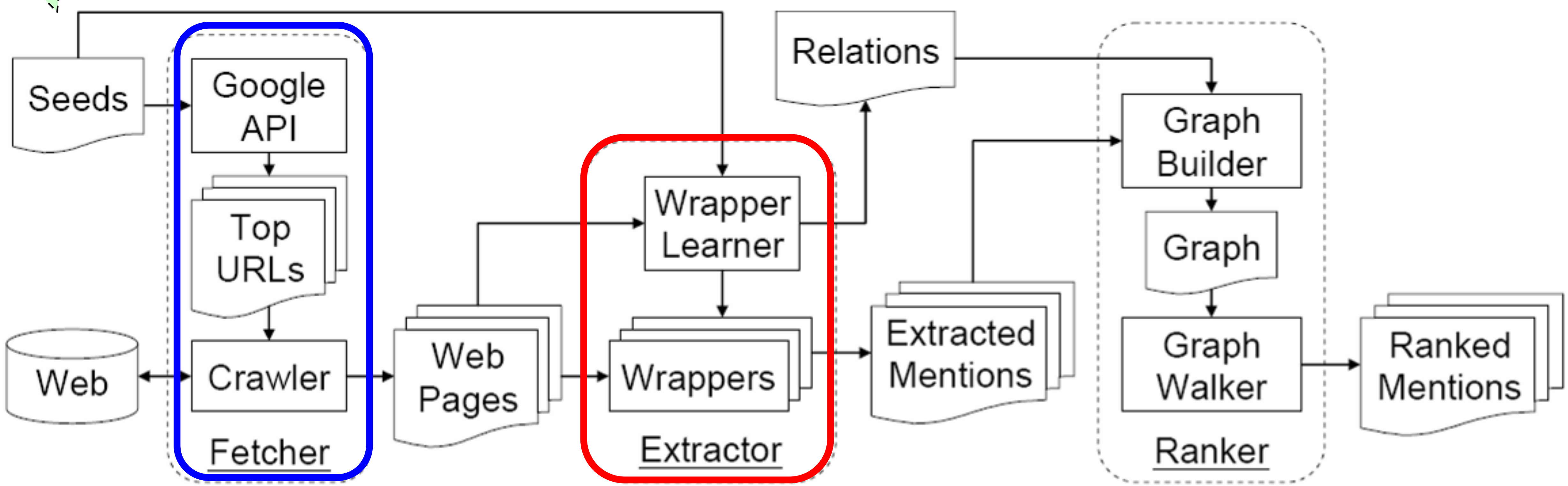


Figure 1. Flow chart of the SEAL system

- **Fetcher**: download web pages from the Web
- **Extractor**: learn wrappers from web pages
- **Ranker**: rank entities extracted by wrappers

SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

1. Canon
2. Nikon
3. Olympus

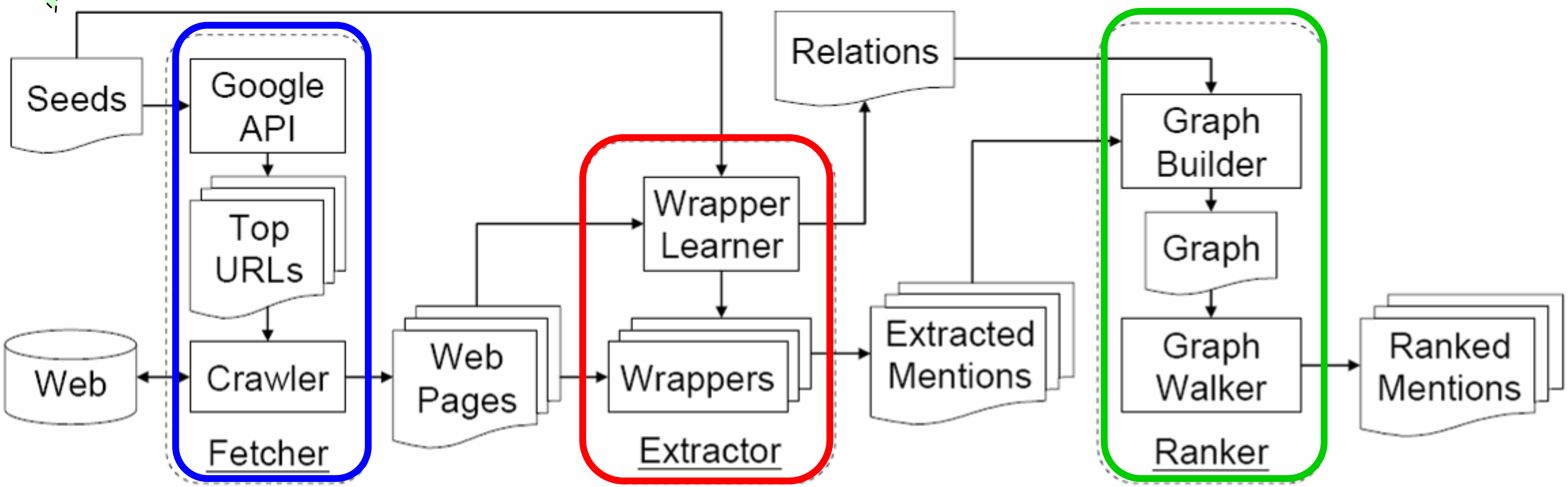


Figure 1. Flow chart of the SEAL system

- **Fetcher**: download web pages from the Web
- **Extractor**: learn wrappers from web pages
- **Ranker**: rank entities extracted by wrappers

SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

1. Canon
2. Nikon
3. Olympus

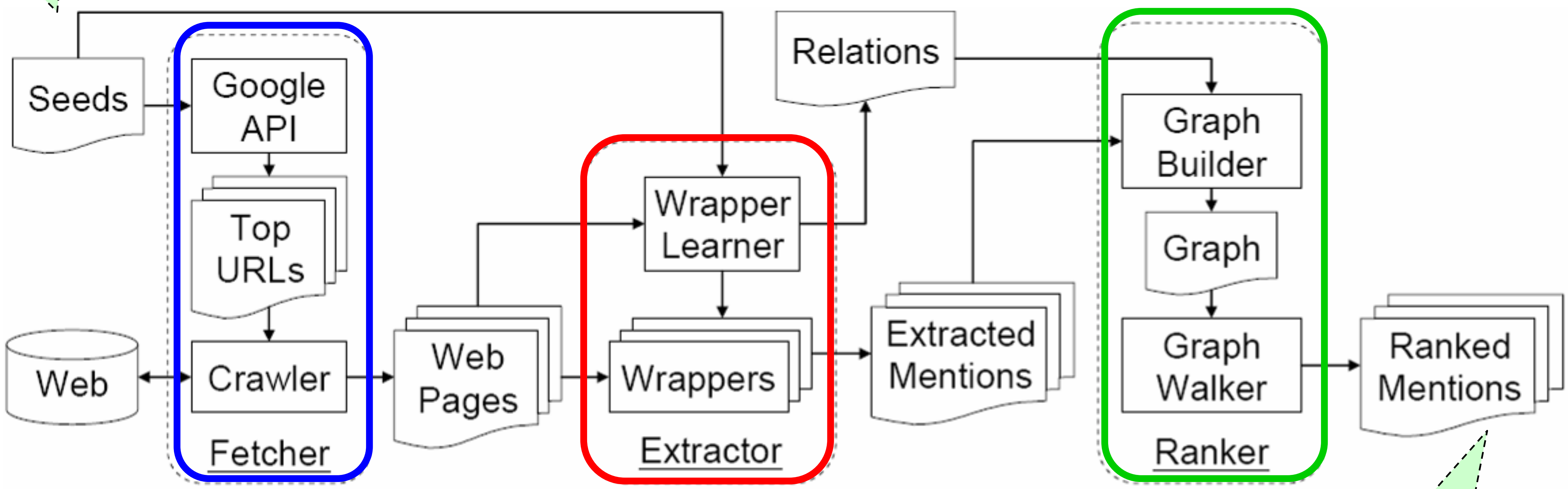


Figure 1. Flow chart of the SEAL system

4. Pentax
5. Sony
6. Kodak
7. Minolta
8. Panasonic
9. Casio
10. Leica
11. Fuji
12. Samsung
13. …

- **Fetcher**: download web pages from the Web
- **Extractor**: learn wrappers from web pages
- **Ranker**: rank entities extracted by wrappers

Ranking Extractions

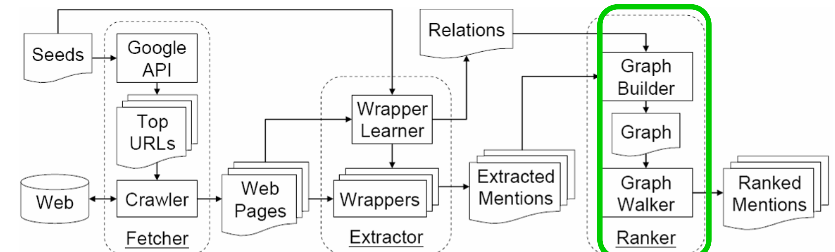


Figure 1. Flow chart of the SEAL system

Table 2. Node and relation types

Source Type	Edge Relation	Target Type
seeds	find	document
document	derive	wrapper
	$find^{-1}$	seeds
wrapper	extract	mention
	$derive^{-1}$	document
mention	$extract^{-1}$	wrapper

- A graph consists of a fixed set of…
  - Node Types: {**seeds**, **document**, **wrapper**, **mention**}
  - Labeled Directed Edges: {**find**, **derive**, **extract**}
    - Each edge asserts that a binary relation r holds
    - Each edge has an inverse relation r^{-1} (graph is cyclic)

Minkov et al. *Contextual Search and Name Disambiguation in Email using Graphs*. SIGIR 2006

Ranking Extractions

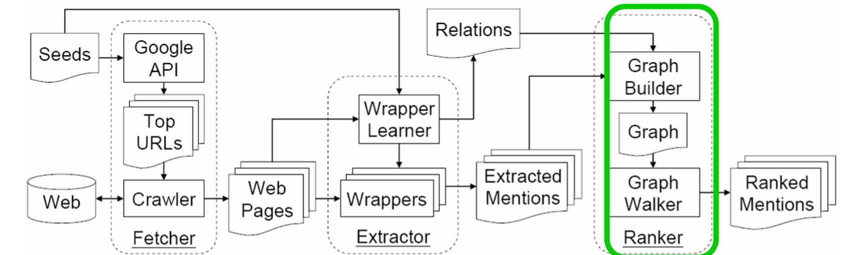
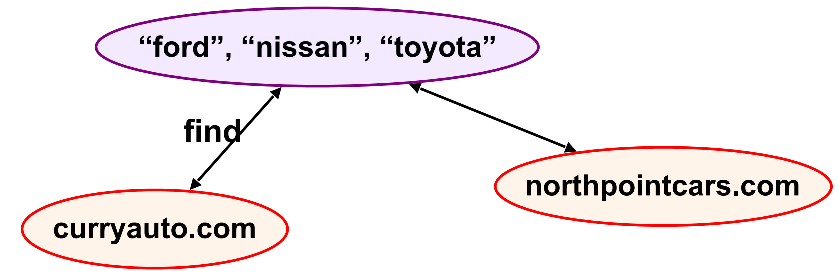


Figure 1. Flow chart of the SEAL system

Table 2. Node and relation types

Source Type	Edge Relation	Target Type
seeds	find	document
document	derive find^{-1}	wrapper seeds
wrapper	extract derive^{-1}	mention document
mention	extract^{-1}	wrapper



- A graph consists of a fixed set of…
  - Node Types: {**seeds**, **document**, **wrapper**, **mention**}
  - Labeled Directed Edges: {**find**, **derive**, **extract**}
    - Each edge asserts that a binary relation r holds
    - Each edge has an inverse relation r^{-1} (graph is cyclic)

Minkov et al. *Contextual Search and Name Disambiguation in Email using Graphs*. SIGIR 2006

Ranking Extractions

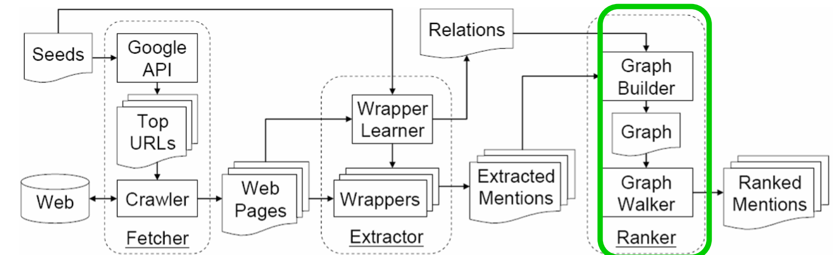
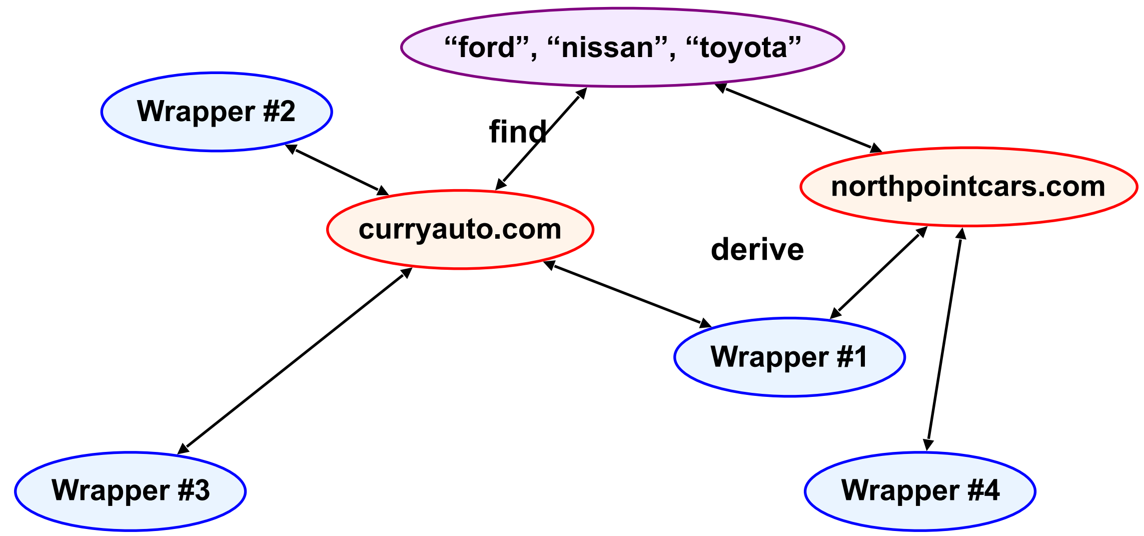


Figure 1. Flow chart of the SEAL system

Table 2. Node and relation types

Source Type	Edge Relation	Target Type
seeds	find	document
document	derive $find^{-1}$	wrapper seeds
wrapper	extract $derive^{-1}$	mention document
mention	$extract^{-1}$	wrapper



- A graph consists of a fixed set of…
  - Node Types: {**seeds**, **document**, **wrapper**, **mention**}
  - Labeled Directed Edges: {**find**, **derive**, **extract**}
    - Each edge asserts that a binary relation r holds
    - Each edge has an inverse relation r^{-1} (graph is cyclic)

Minkov et al. *Contextual Search and Name Disambiguation in Email using Graphs*. SIGIR 2006

Ranking Extractions

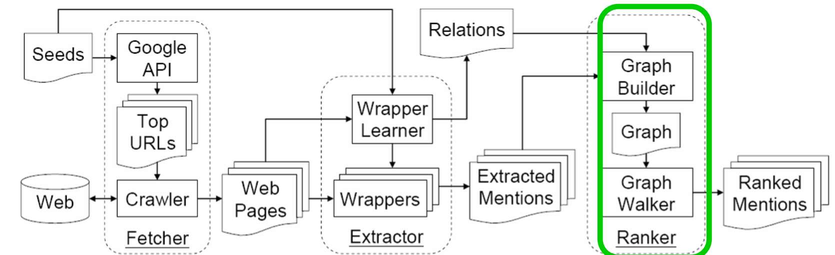
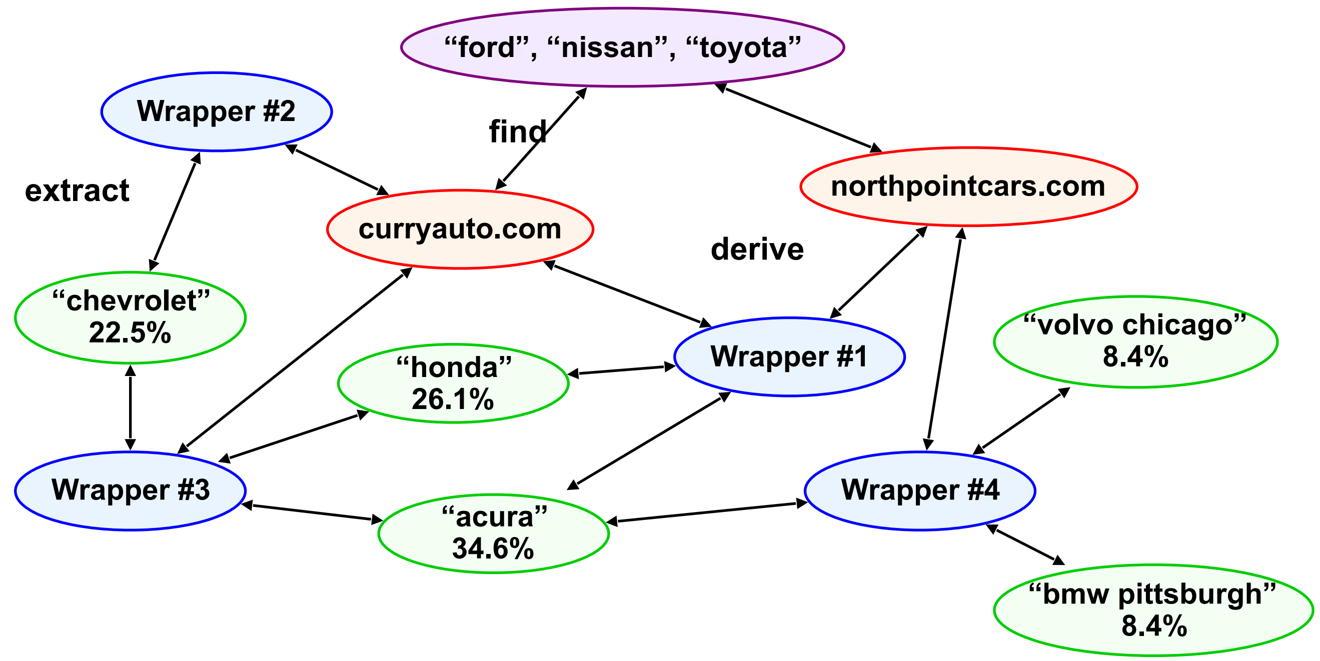


Figure 1. Flow chart of the SEAL system

Table 2. Node and relation types

Source Type	Edge Relation	Target Type
seeds	find	document
document	derive $find^{-1}$	wrapper seeds
wrapper	extract $derive^{-1}$	mention document
mention	$extract^{-1}$	wrapper



- A graph consists of a fixed set of…
  - Node Types: {**seeds**, **document**, **wrapper**, **mention**}
  - Labeled Directed Edges: {**find**, **derive**, **extract**}
    - Each edge asserts that a binary relation r holds
    - Each edge has an inverse relation r^{-1} (graph is cyclic)

Minkov et al. *Contextual Search and Name Disambiguation in Email using Graphs*. SIGIR 2006

Ranking Extractions

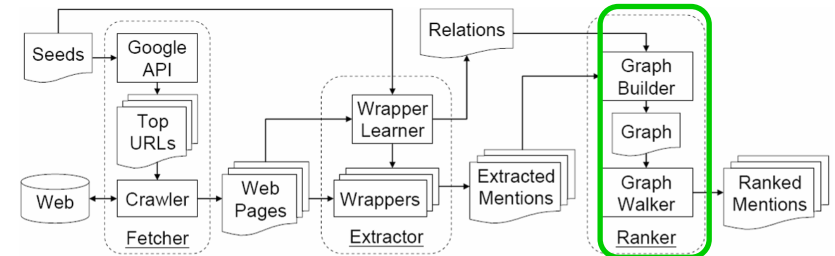
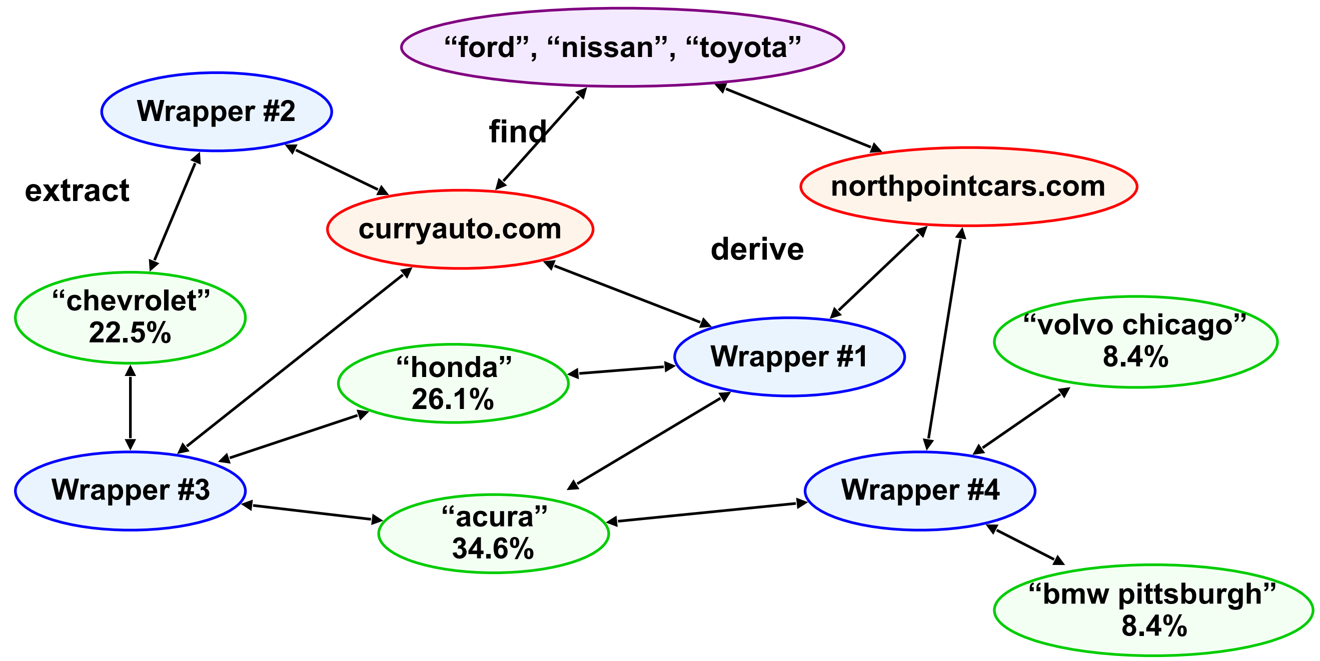


Figure 1. Flow chart of the SEAL system

Table 2. Node and relation types

Source Type	Edge Relation	Target Type
seeds	find	document
document	derive	wrapper
	$find^{-1}$	seeds
wrapper	extract	mention
	$derive^{-1}$	document
mention	$extract^{-1}$	wrapper



- A graph consists of a fixed set of…
  - Node Types: {**seeds**, **document**, **wrapper**, **mention**}
  - Labeled Directed Edges: {**find**, **derive**, **extract**}
    - Each edge asserts that a binary relation r holds
    - Each edge has an inverse relation r^{-1} (graph is cyclic)

Minkov et al. *Contextual Search and Name Disambiguation in Email using Graphs*. SIGIR 2006

Top three mentions are the seeds

#	Entity	#	Entity	#	Entity	#	Entity	#	Entity	#	Entity	#	Entity
1	kdd	1	andrew mccallum	1	survivor	1	sam's club	1	梅花	1	睡美人	1	ドラえもん
2	icml	2	michael collins	2	amazing race	2	walmart	2	牡丹花	2	灰姑娘	2	ハローキティ
3	icdm	3	john lafferty	3	american idol	3	home depot	3	杜鵑花	3	白雪公主	3	ポケモン
4	ijcai	4	naftali tishby	4	big brother	4	target	4	蘭花	4	小紅帽	4	スヌーピー
5	aaai	5	fernando pereira	5	the apprentice	5	sears	5	茉莉花	5	美人魚	5	くまのプーさん
6	ecml	6	zoubin ghahramani	6	the bachelor	6	circuit city	6	月季花	6	小美人魚	6	アンパンマン
7	nips	7	daphne koller	7	the mole	7	best buy	7	梔子花	7	美女與野獸	7	ムーミン
8	sdm	8	thomas hofmann	8	joe millionaire	8	ace hardware	8	菊花	8	花木蘭	8	ワンピース
9	pkdd	9	thorsten joachims	9	average joe	9	office depot	9	瓊花	9	青蛙王子	9	シナモロール
10	sigir	10	david heckerman	10	reality tv	10	kmart	10	桃花	10	貝兒	10	ケロロ軍曹
11	pakdd	11	nir friedman	11	nashville star	11	drugstore.com	11	玉蘭花	11	木偶奇遇記	11	ミッキーマウス
12	colt	12	tom mitchell	12	dancing with the stars	12	sephora	12	海棠花	12	糖果屋	12	リラックマ
13	cikm	13	dan roth	13	surreal life	13	the sports authority	13	水仙花	13	三隻小豬	13	ピングー
14	ida	14	william w. cohen	14	the bachelorette	14	staples	14	桂花	14	茉莉公主	14	ピーターラビット
15	uai	15	mark craven	15	road rules	15	blockbuster	15	杏花	15	茉莉	15	ミッフィー
16	ilp	16	roni rosenfeld	16	fear factor	16	rei	16	合歡花	16	愛麗絲夢遊仙境	16	トトロ
17	stoc	17	david mcallester	17	paradise hotel	17	toys r us	17	繡球花	17	寶嘉康蒂	17	マイメロディ
18	www	18	yoram singer	18	america's next top model	18	nordstrom	18	櫻花	18	長髮姑娘	18	機関車トーマス
19	alt	19	michael i. jordan	19	lost	19	dick's sporting goods	19	虞美人花	19	人魚公主	19	セサミストリート
20	icde	20	eugene charniak	20	joe schmo	20	lowes	20	青黛花	20	紅舞鞋	20	ウルトラマン
21	sigmod	21	amir globerson	21	extreme makeover	21	aafes	21	十姊妹花	21	唐老鴨	21	ディズニー
22	ecai	22	yiming yang	22	temptation	22	fred meyer	22	木棉花	22	長靴貓	22	恐竜キング
23	dawak	23	yoshua bengio	23	celebrity mole	23	orchard supply	23	眞珠蘭花	23	拇指神童	23	ムシキング
24	cvpr	24	sridhar mahadevan	24	desperate housewives	24	handy hardware	24	楊花	24	小熊維尼	24	おじゃる丸

Extraction Techniques

Extraction Techniques

....
What Other Musicians Would Fans of the Album Listen to:
Storytelling musicians come to mind. Musicians *such as* Johnny Cash, and Woodie Guthrie.

What is Distinctive About this Release?:
Every song on the album has its own unique sound. From the fast paced *That Texas Girl* to the acoustic ....

[van Durme and Pasca, AAAI 2008]

- Uses "<Class> *such as* <Instance>" patterns
- Extracts both class (musician) and instance (Johnny Cash)

Extraction Techniques

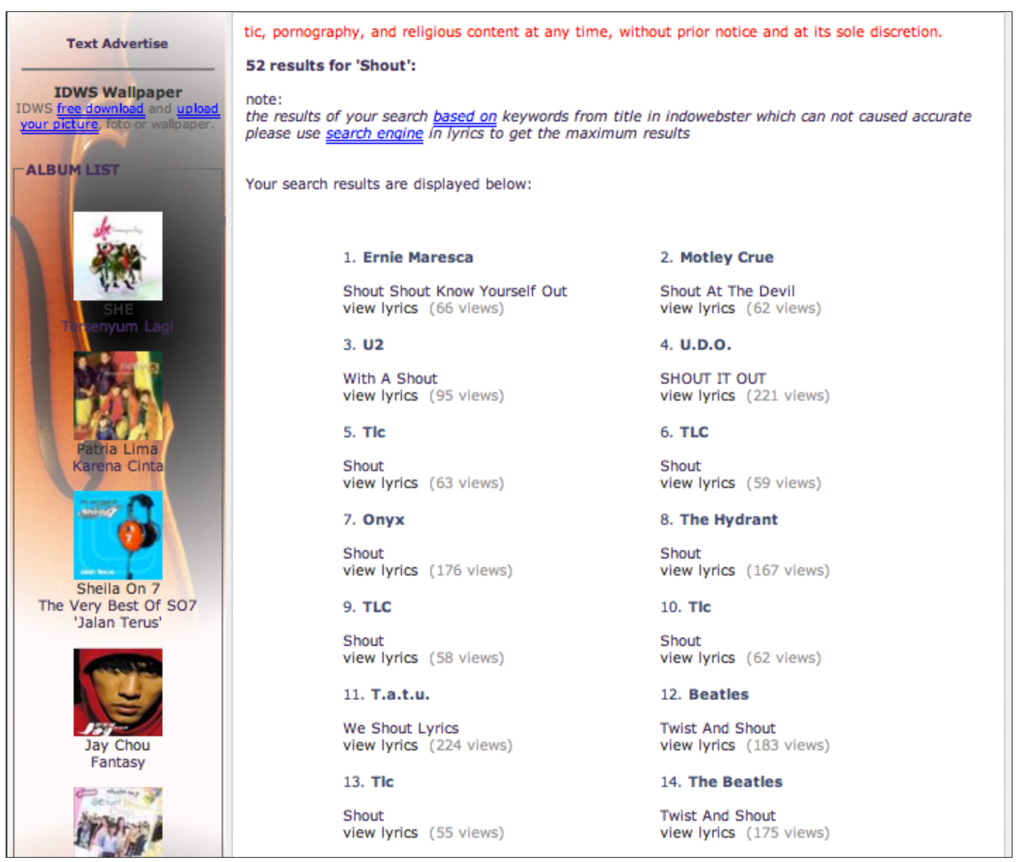
....
What Other Musicians Would Fans of the Album Listen to:

Storytelling musicians come to mind. Musicians *such as* Johnny Cash, and Woodie Guthrie.

What is Distinctive About this Release?:
Every song on the album has its own unique sound. From the fast paced *That Texas Girl* to the acoustic ....

[van Durme and Pasca, AAAI 2008]

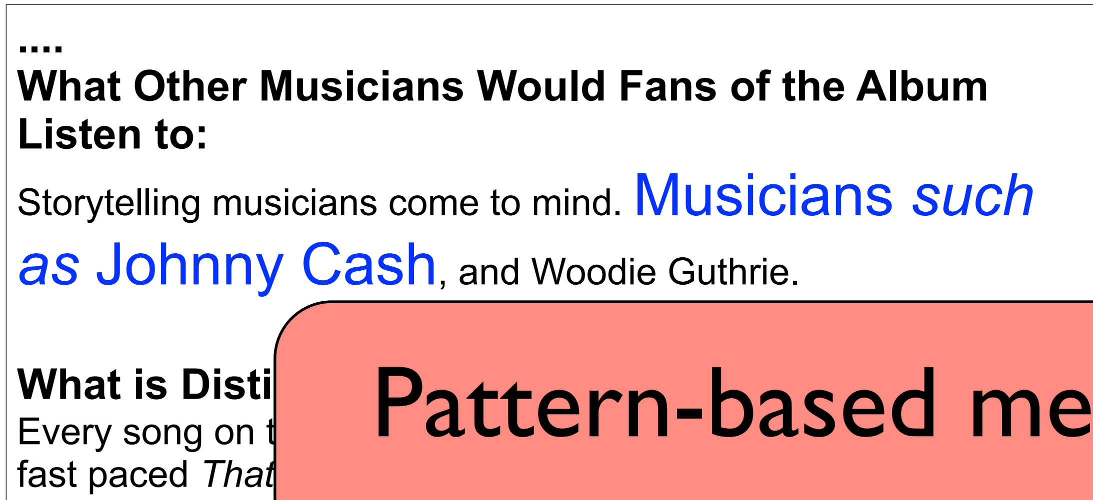
- Uses "<Class> *such as* <Instance>" patterns
- Extracts both class (musician) and instance (Johnny Cash)



Extractions from HTML lists and tables

- SEAL [Wang and Cohen, ICDM 2007]
- WebTables [Cafarella et al., VLDB 2008], 154 million HTML tables

Extraction Techniques



[van Durme and Pasca, AAAI 2008]

- Uses "<Class> *such as* <Instance>"

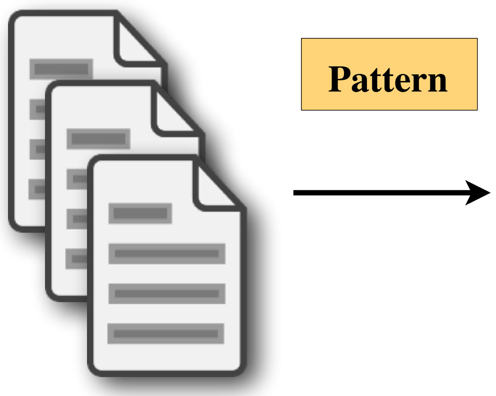
Pattern-based methods are usually tuned for high-precision, resulting in low coverage

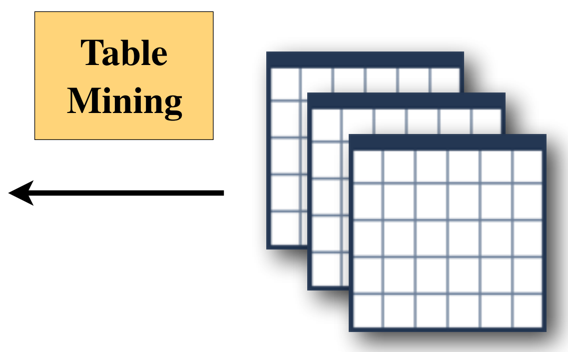
Can we combine extractions from all methods (and sources) to improve coverage?

- SEAL [Wang and Cohen, ICDM 2007]
- WebTables [Cafarella et al., VLDB 2008], 154 million HTML tables

Class-instance Acquisition using Graph-based SSL

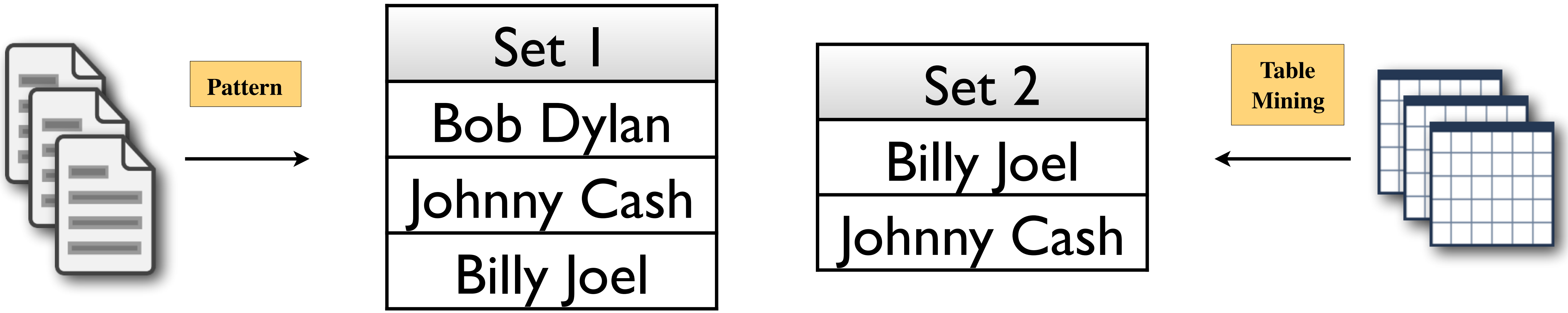
[Talukdar et al., EMNLP 2008, 2010]





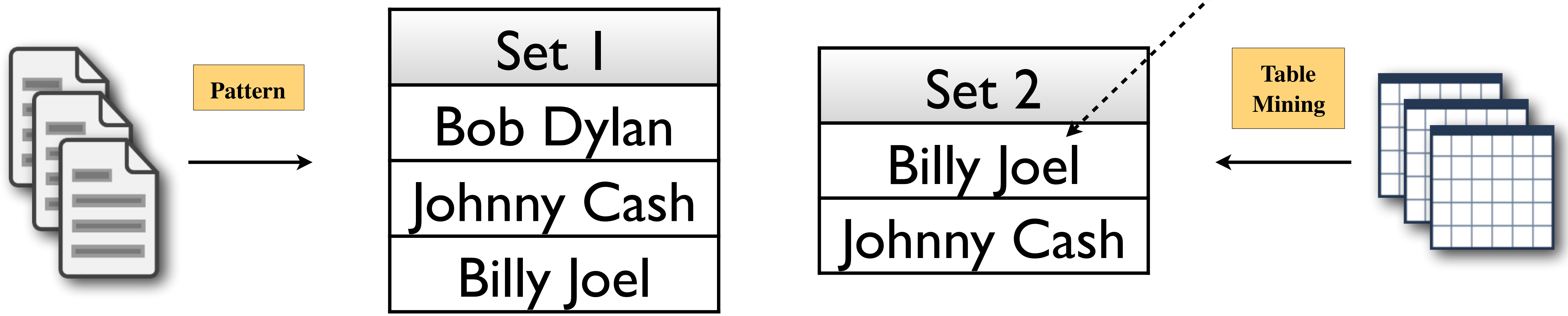
Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]



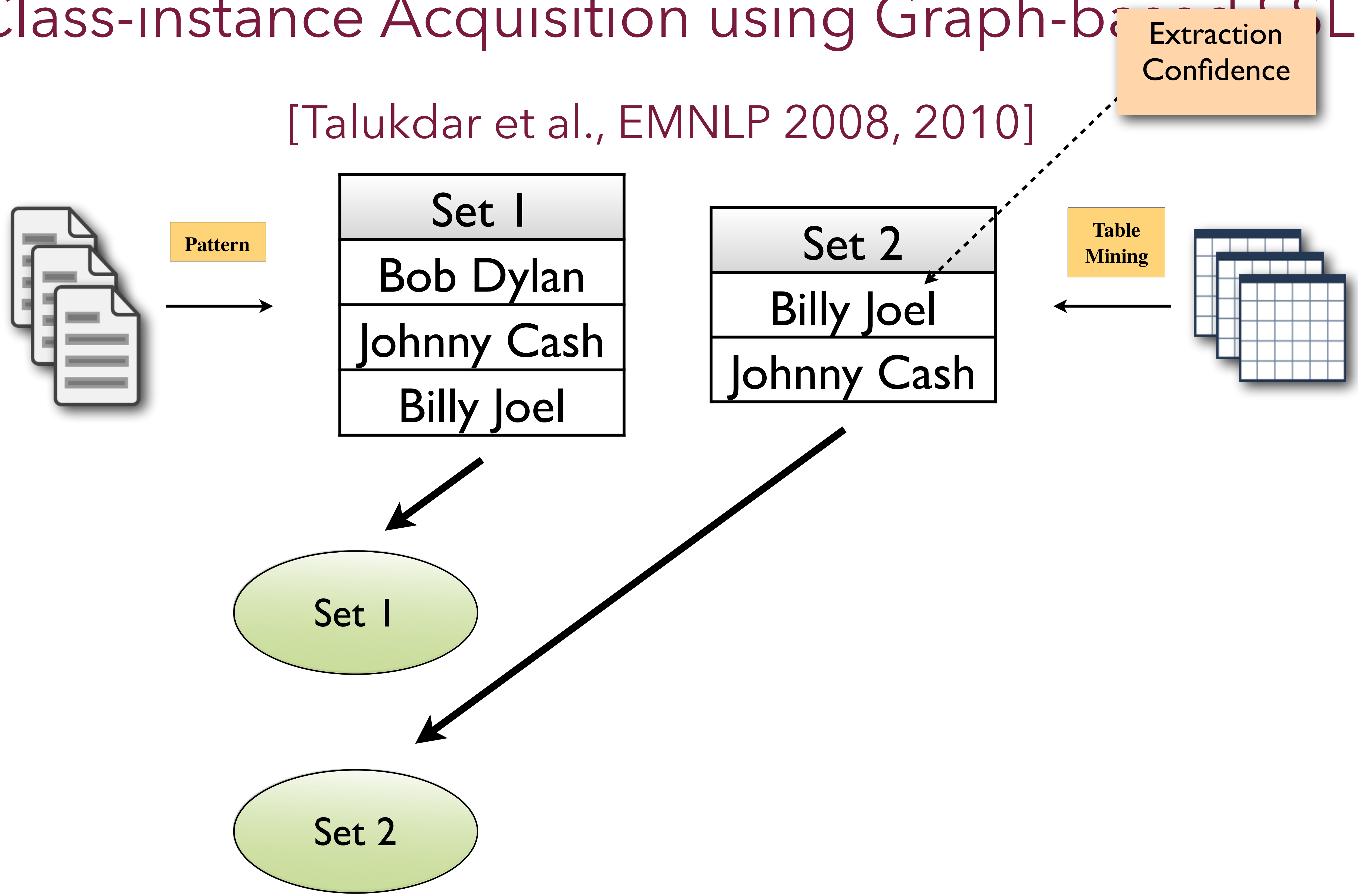
Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]



Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]



Class-instance Acquisition using Graph-based SSL

Extraction Confidence

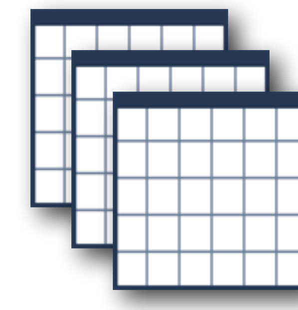
[Talukdar et al., EMNLP 2008, 2010]

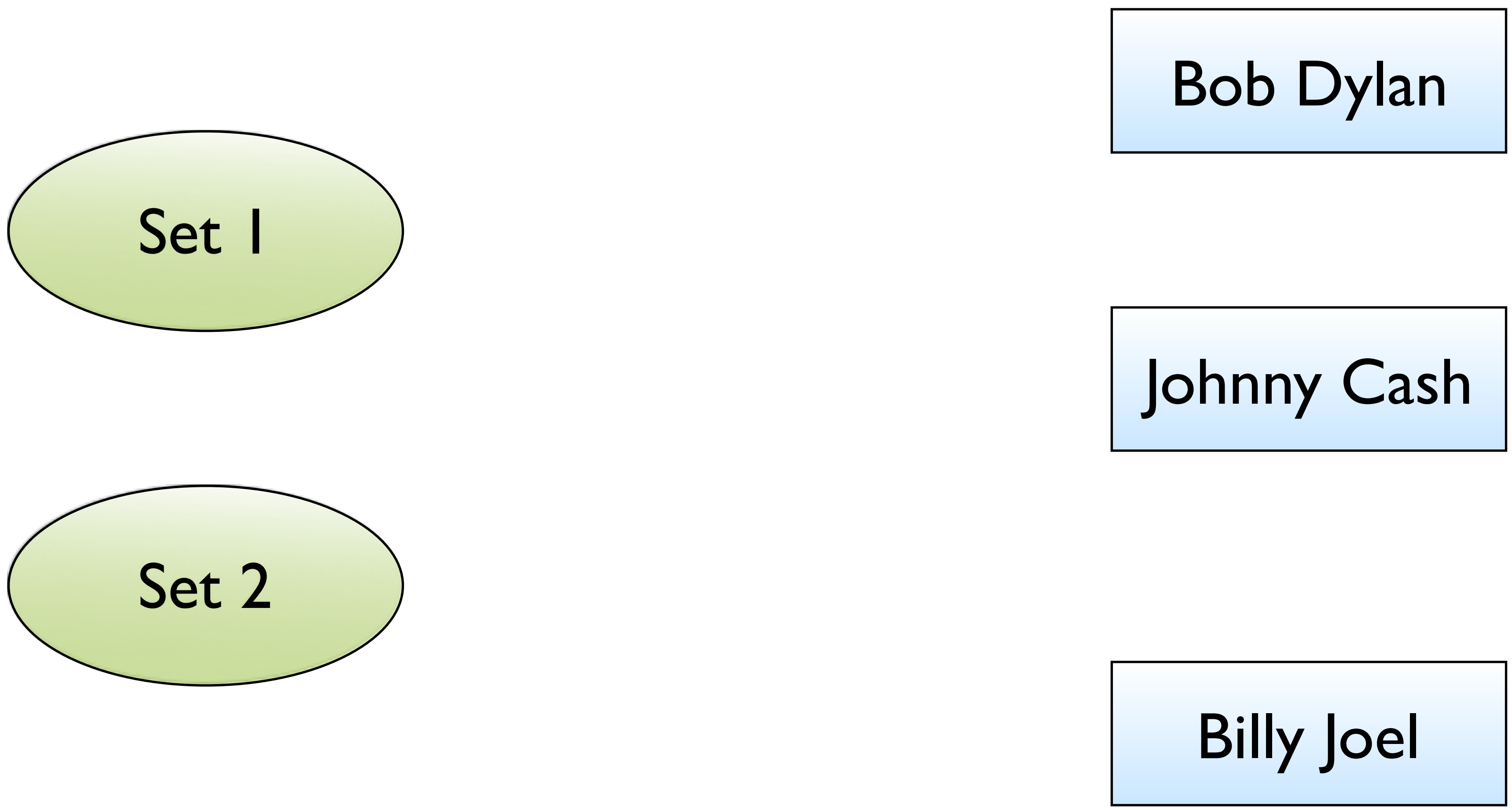
Pattern

Set 1
Bob Dylan
Johnny Cash
Billy Joel

Set 2
Billy Joel
Johnny Cash

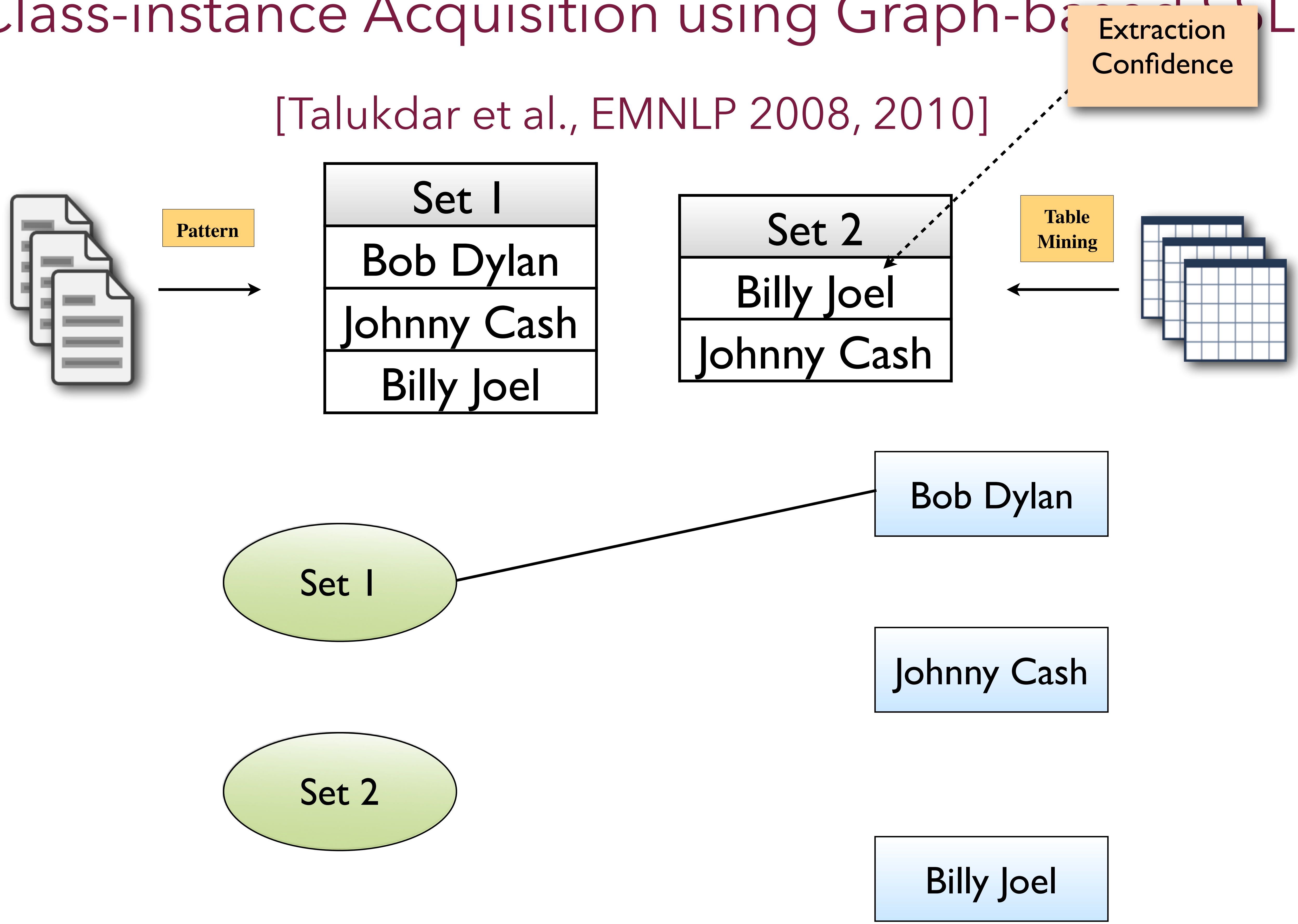
Table Mining





Class-instance Acquisition using Graph-based SSL

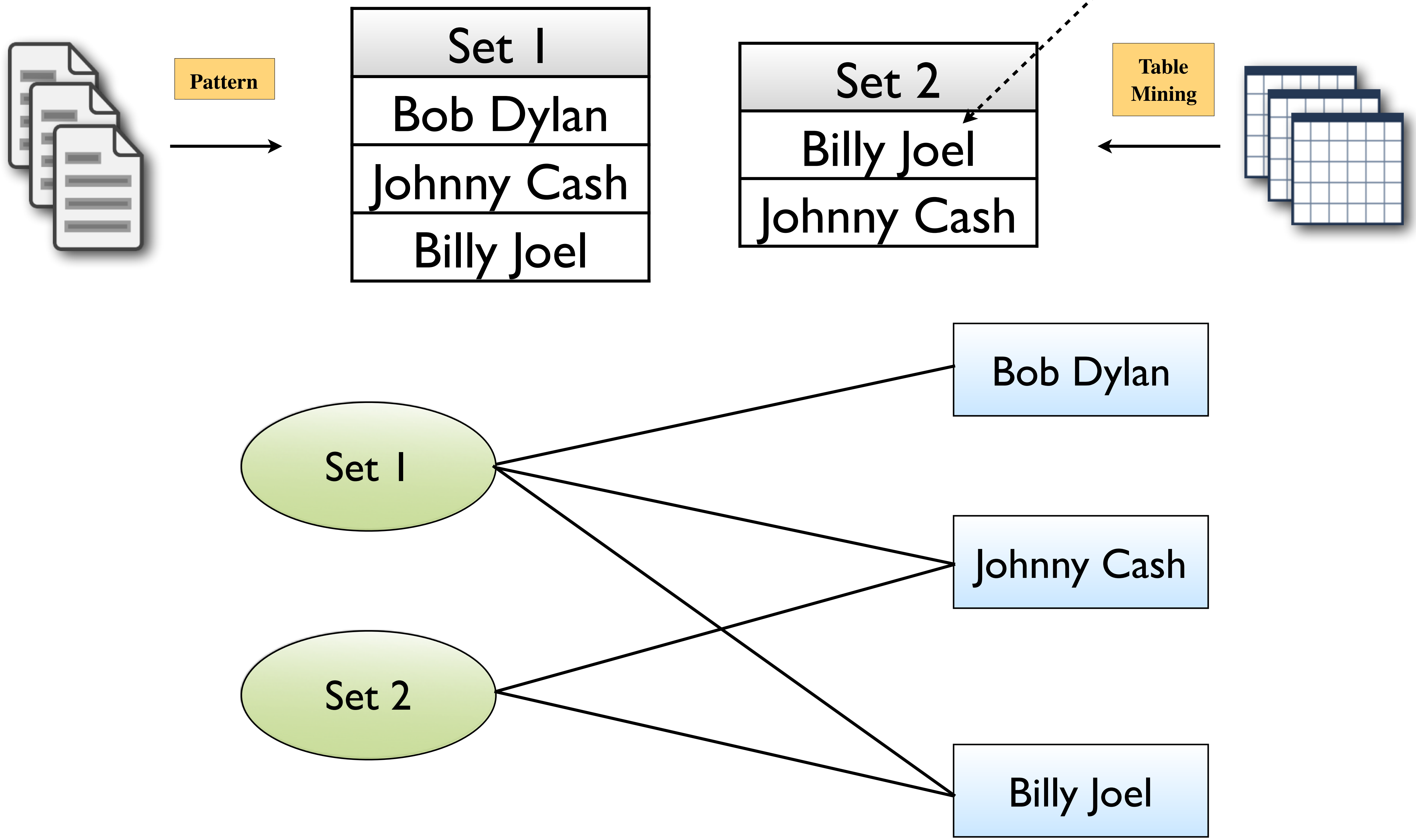
[Talukdar et al., EMNLP 2008, 2010]



Class-instance Acquisition using Graph-based SSL

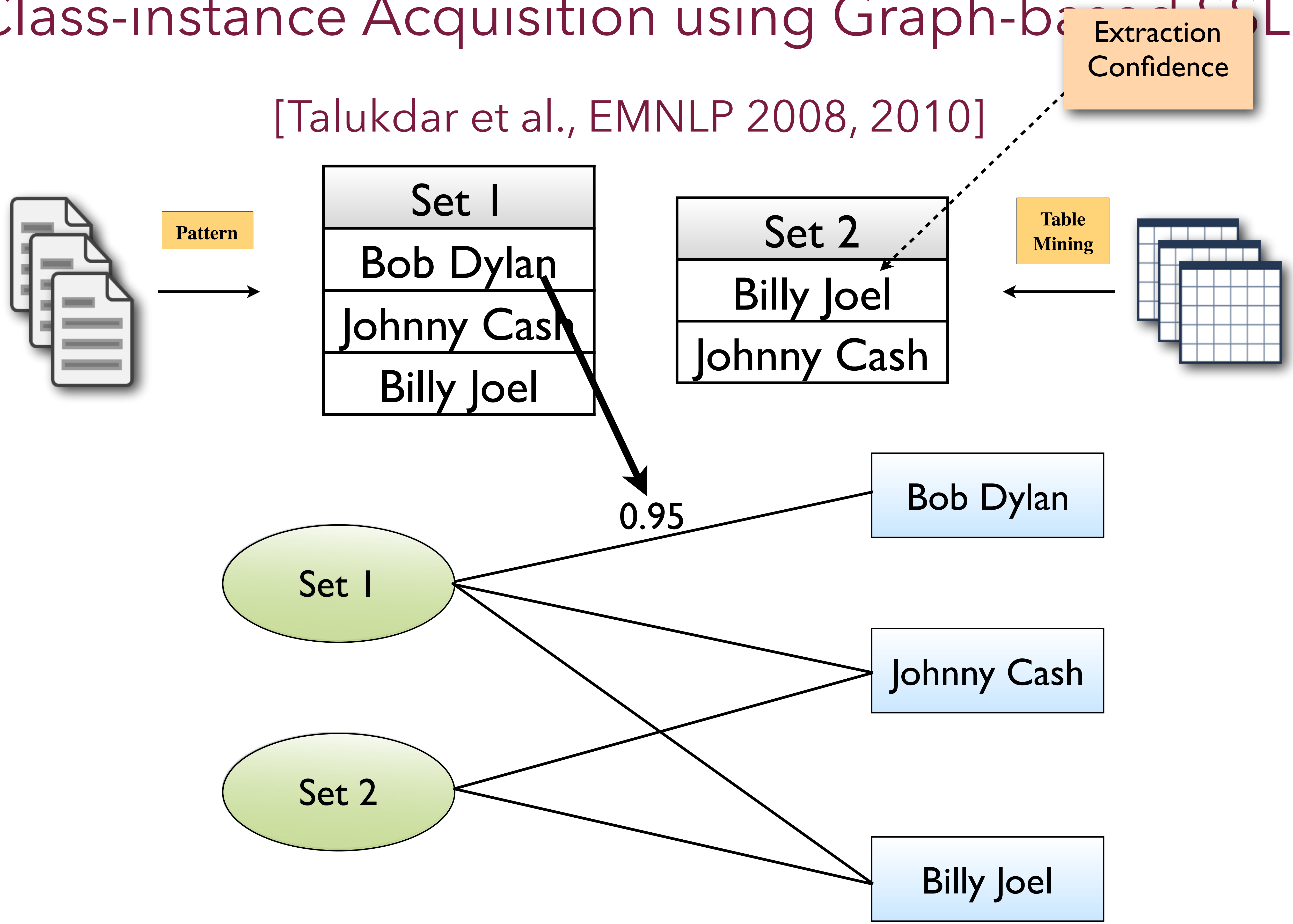
Extraction
Confidence

[Talukdar et al., EMNLP 2008, 2010]



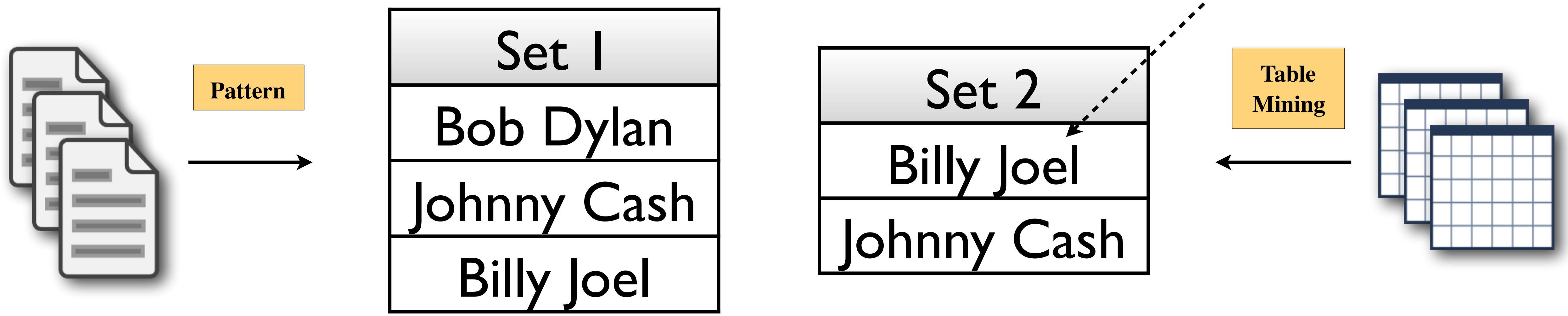
Class-instance Acquisition using Graph-based SSL

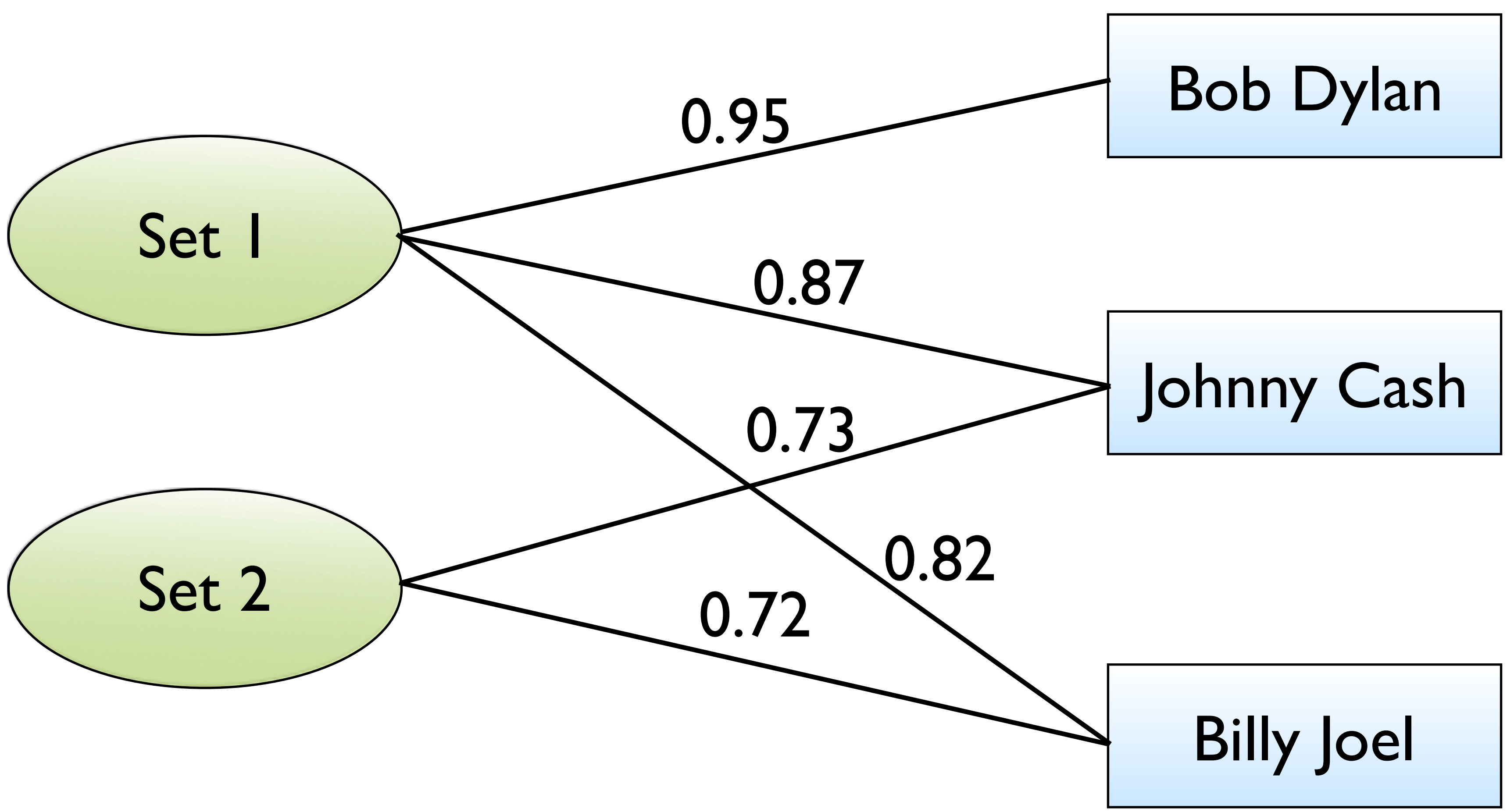
[Talukdar et al., EMNLP 2008, 2010]



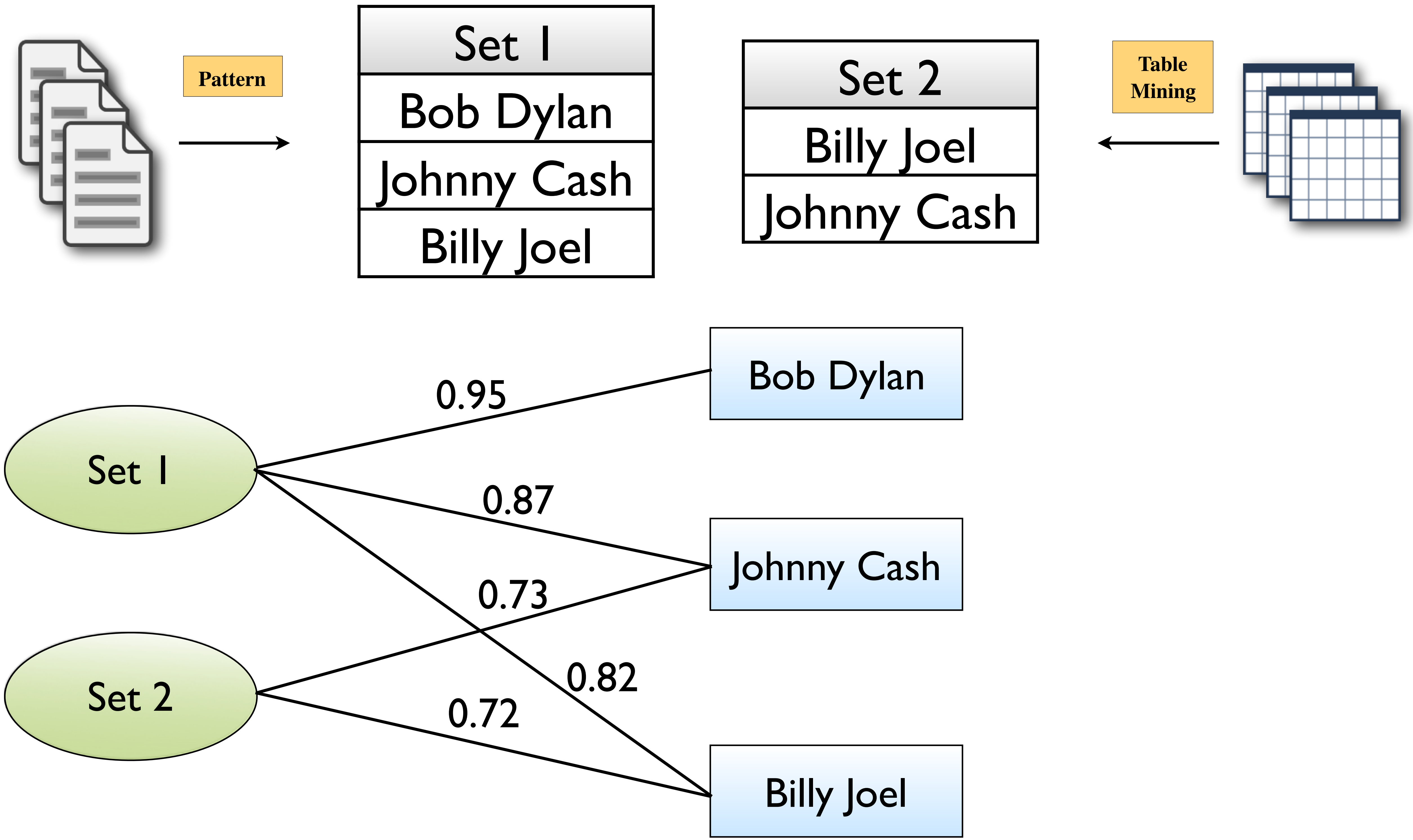
Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]

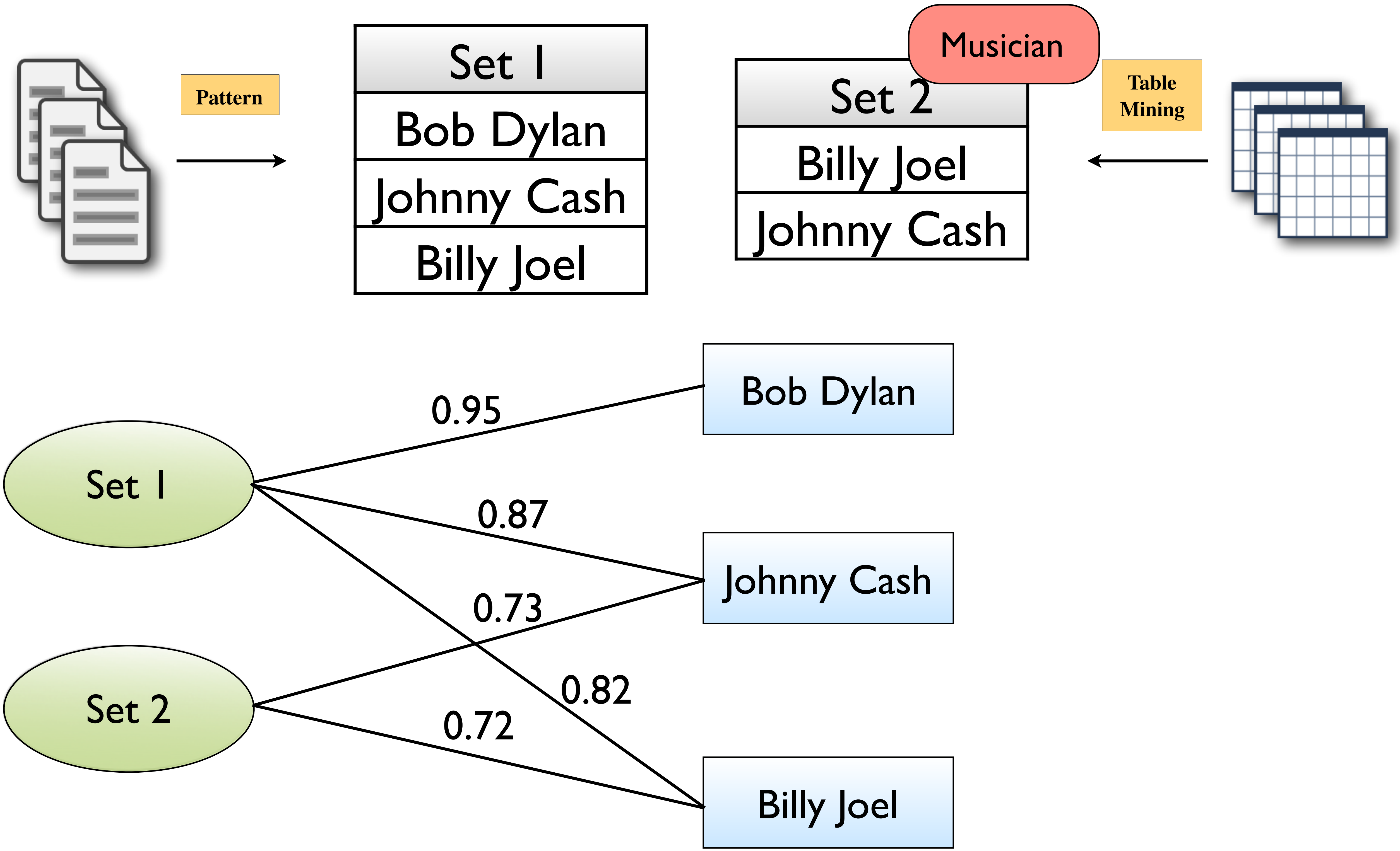




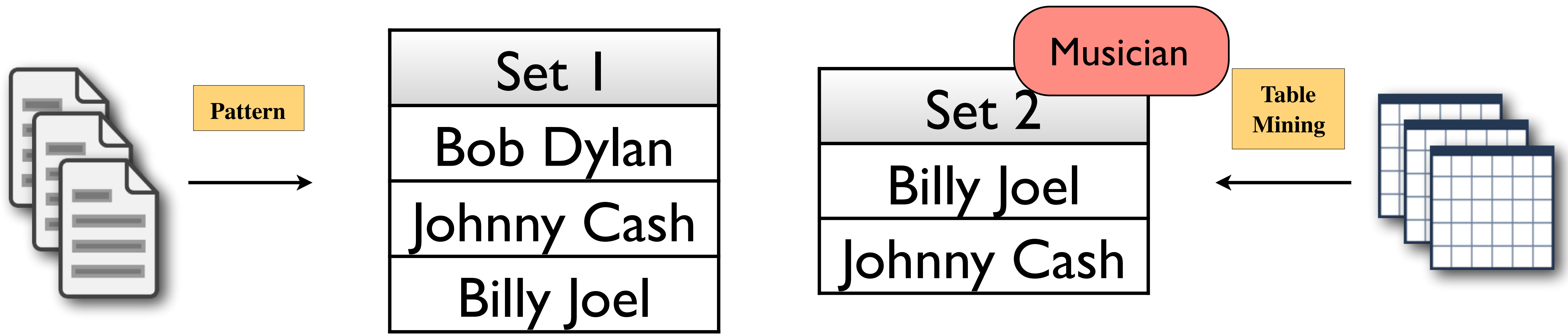
Goal

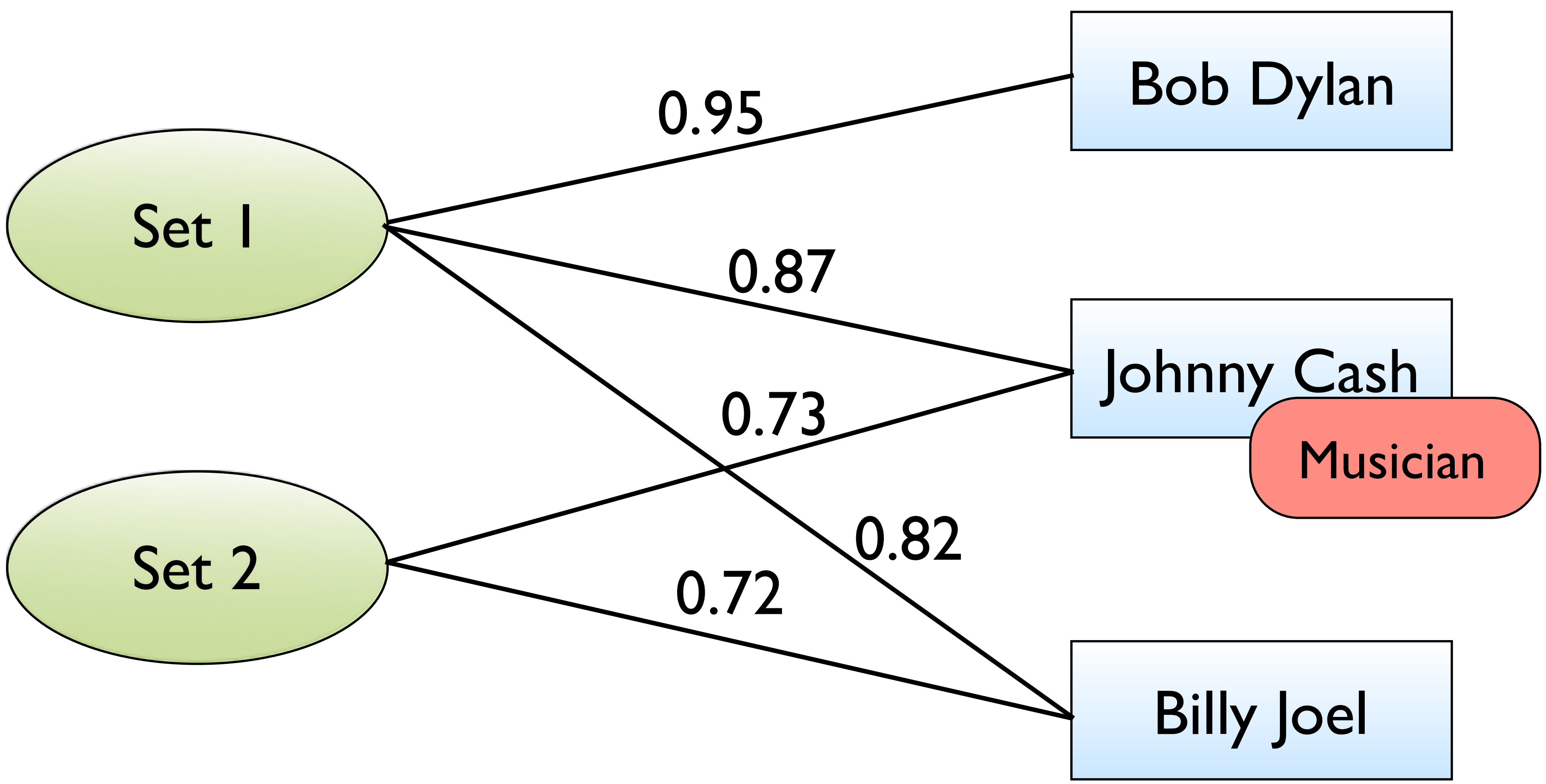

Pattern
Set 1
Bob Dylan
Johnny Cash
Billy Joel
Set 2
Billy Joel
Johnny Cash
Table Mining
Set 1
Set 2
Bob Dylan
Johnny Cash
Billy Joel
0.95
0.87
0.73
0.82
0.72

Goal

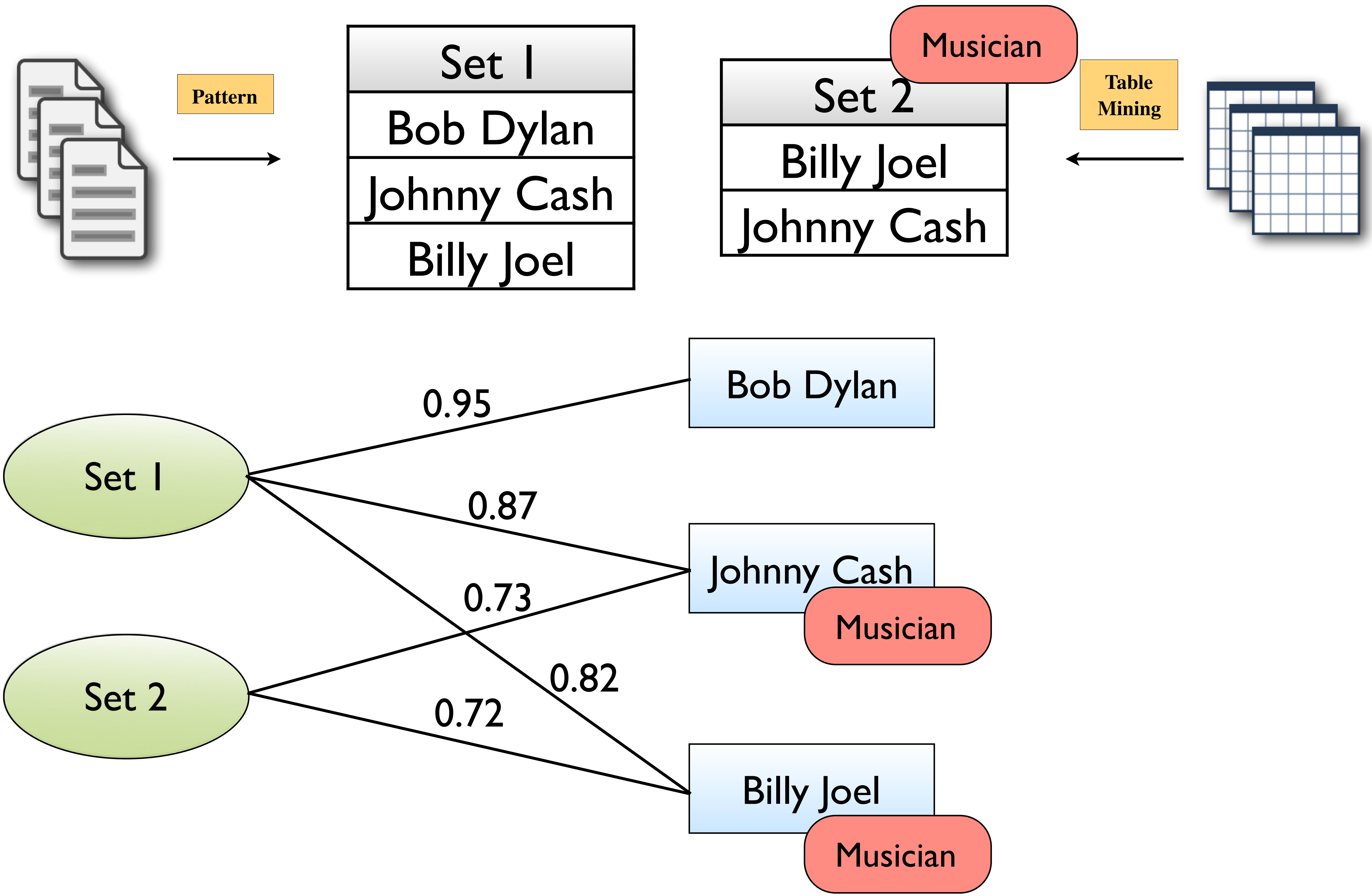

Pattern
Set 1
Bob Dylan
Johnny Cash
Billy Joel
Musician
Set 2
Billy Joel
Johnny Cash
Table Mining
Set 1
Set 2
0.95
0.87
0.73
0.82
0.72
Bob Dylan
Johnny Cash
Billy Joel

Goal

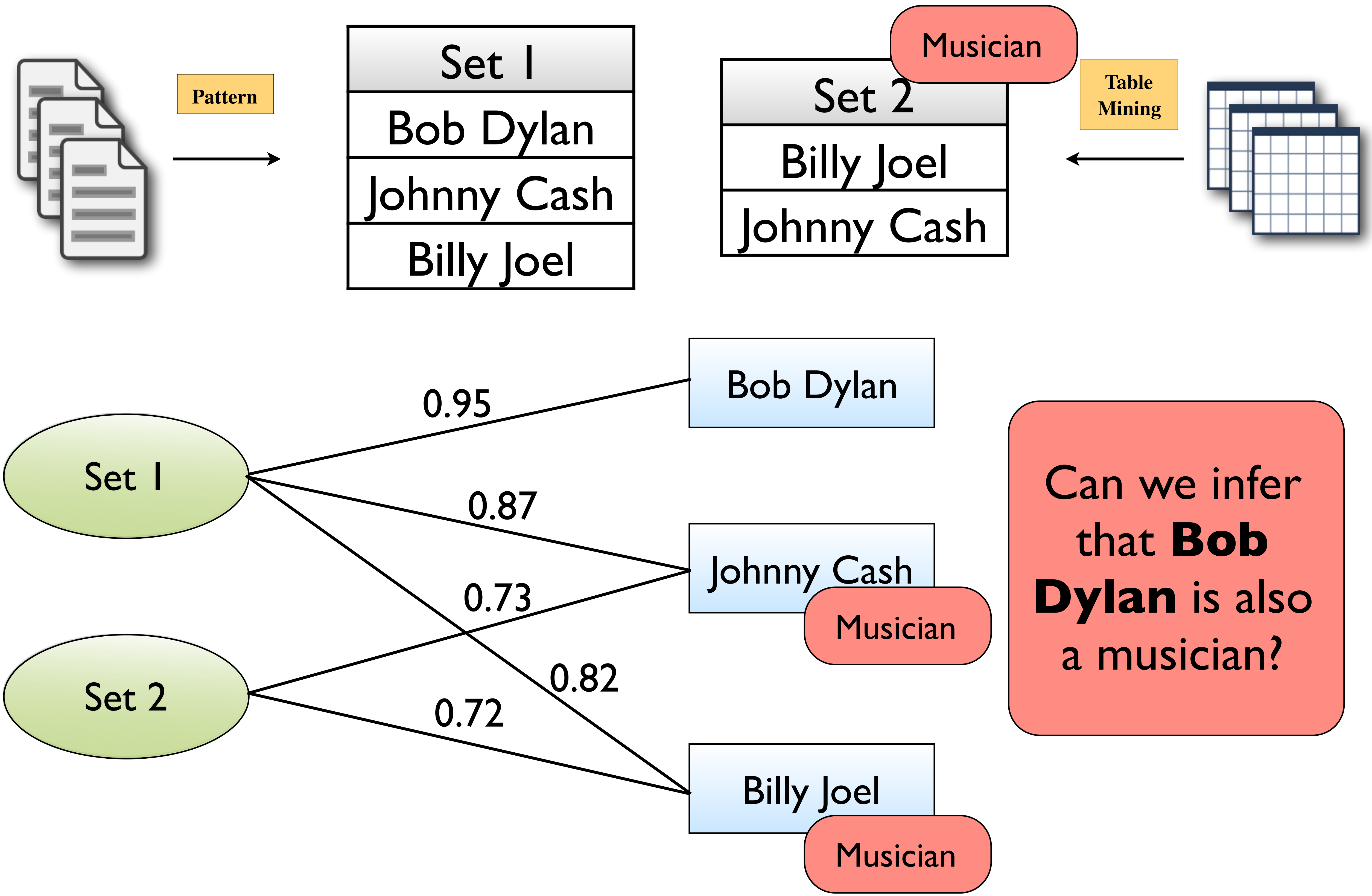

Pattern
Set 1
Bob Dylan
Johnny Cash
Billy Joel
Musician
Set 2
Billy Joel
Johnny Cash
Table Mining


Set 1
Set 2
Bob Dylan
Johnny Cash
Musician
Billy Joel
0.95
0.87
0.73
0.82
0.72

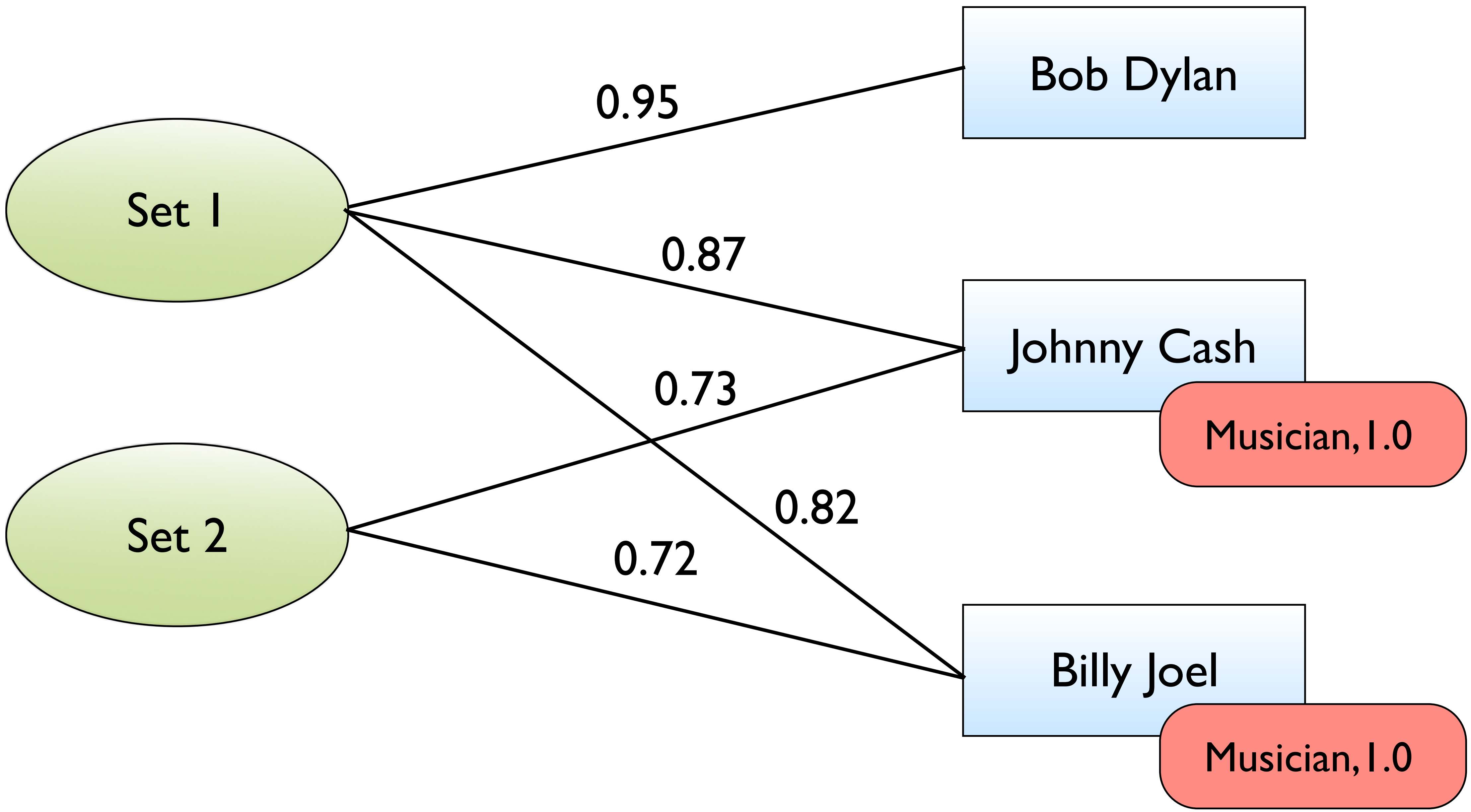
Goal


Pattern
Set 1
Bob Dylan
Johnny Cash
Billy Joel
Musician
Set 2
Billy Joel
Johnny Cash
Table Mining
Set 1
Set 2
0.95
0.87
0.73
0.82
0.72
Bob Dylan
Johnny Cash
Musician
Billy Joel
Musician

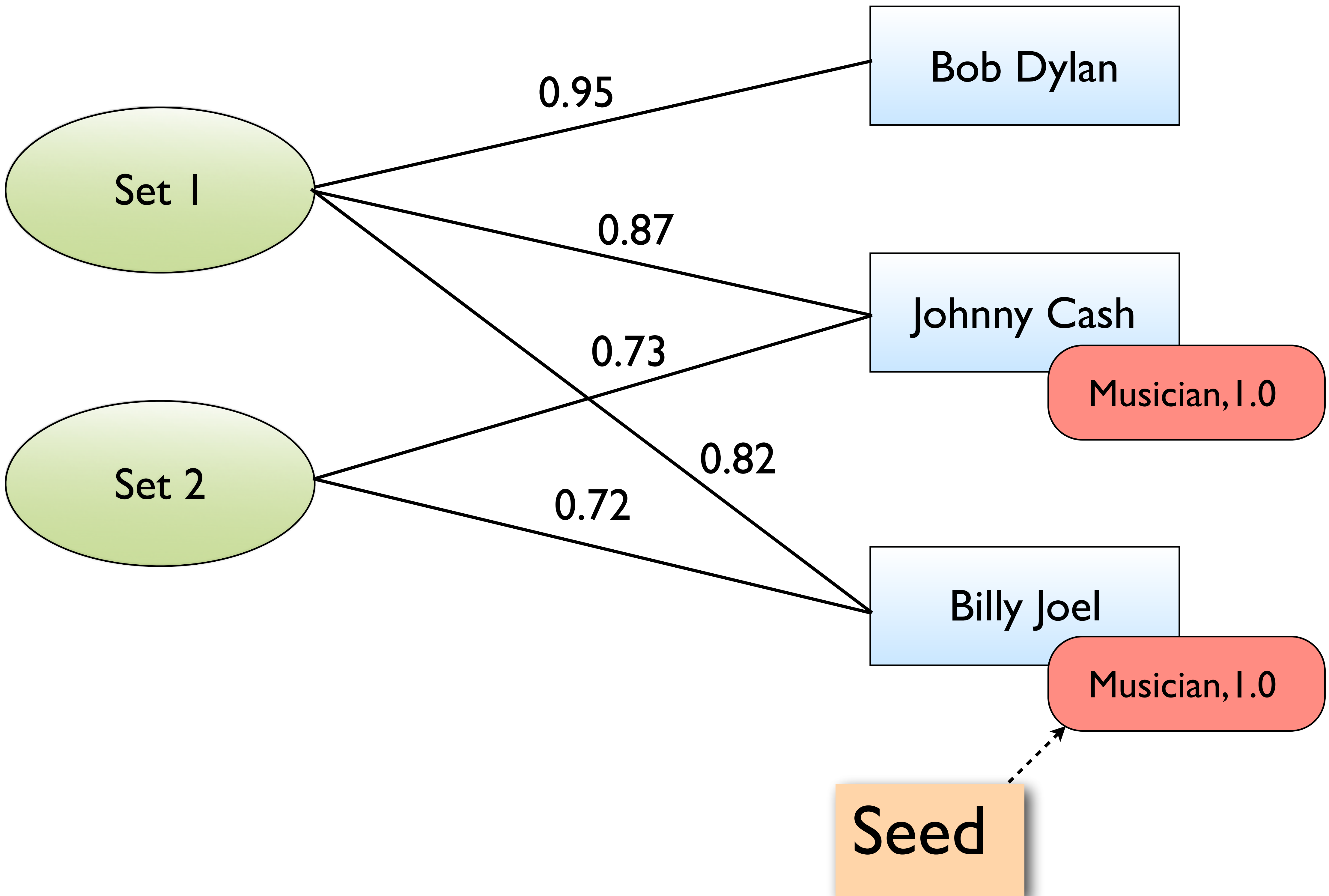
Goal


Pattern
Set 1
Bob Dylan
Johnny Cash
Billy Joel
Musician
Set 2
Billy Joel
Johnny Cash
Table Mining
Set 1
Set 2
0.95
0.87
0.73
0.82
0.72
Bob Dylan
Johnny Cash
Musician
Billy Joel
Musician
Can we infer that **Bob Dylan** is also a musician?

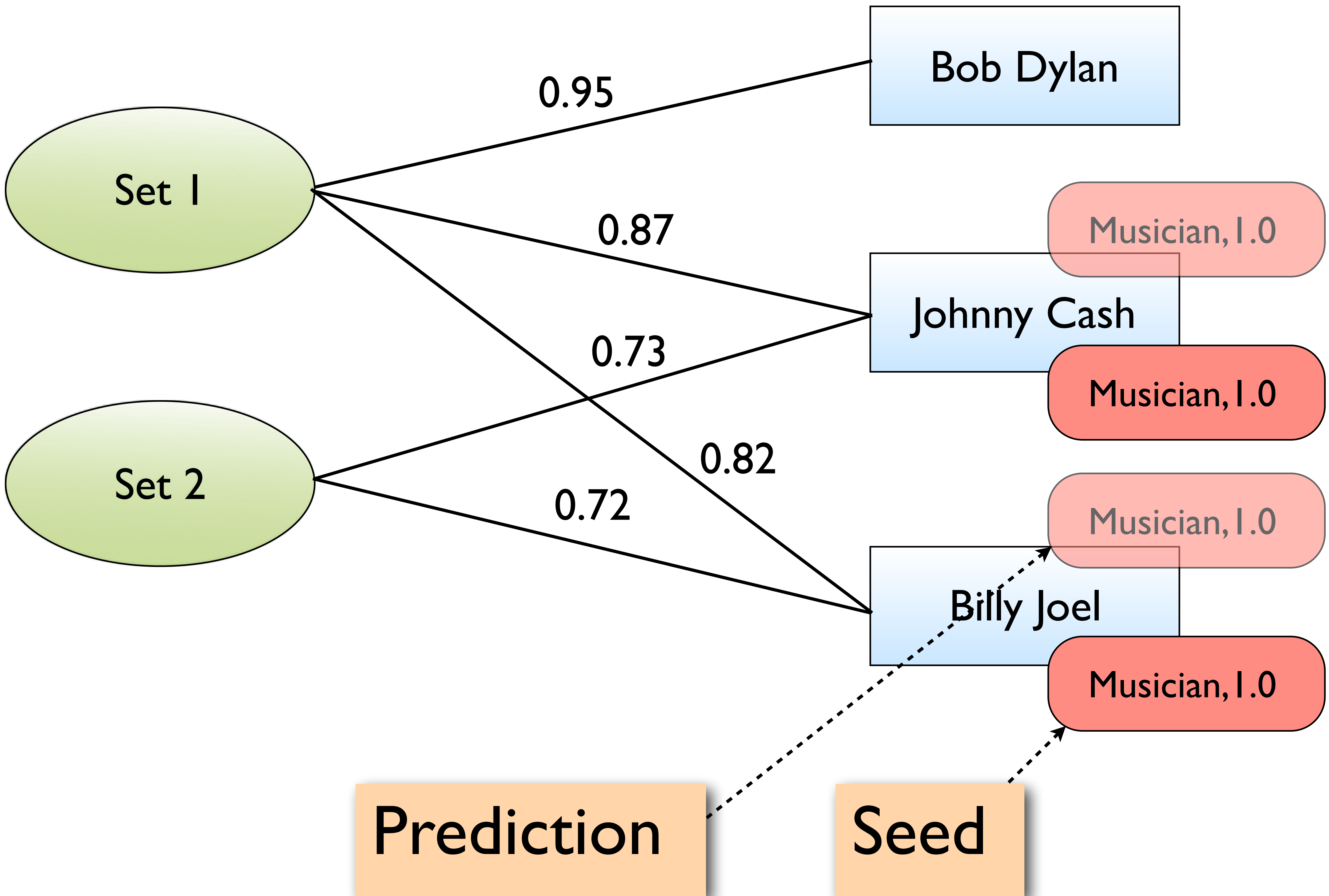
Graph Propagation



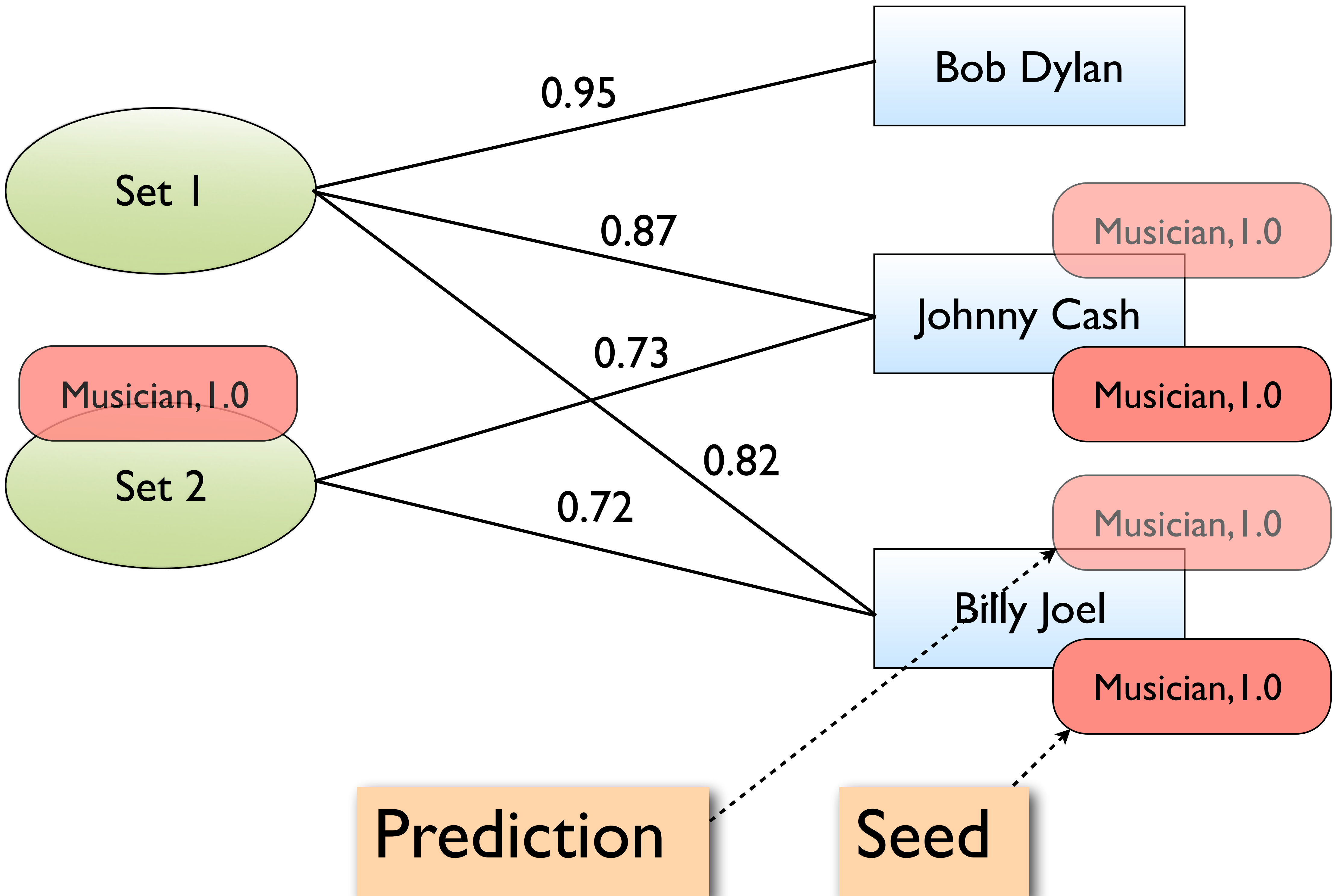
Graph Propagation



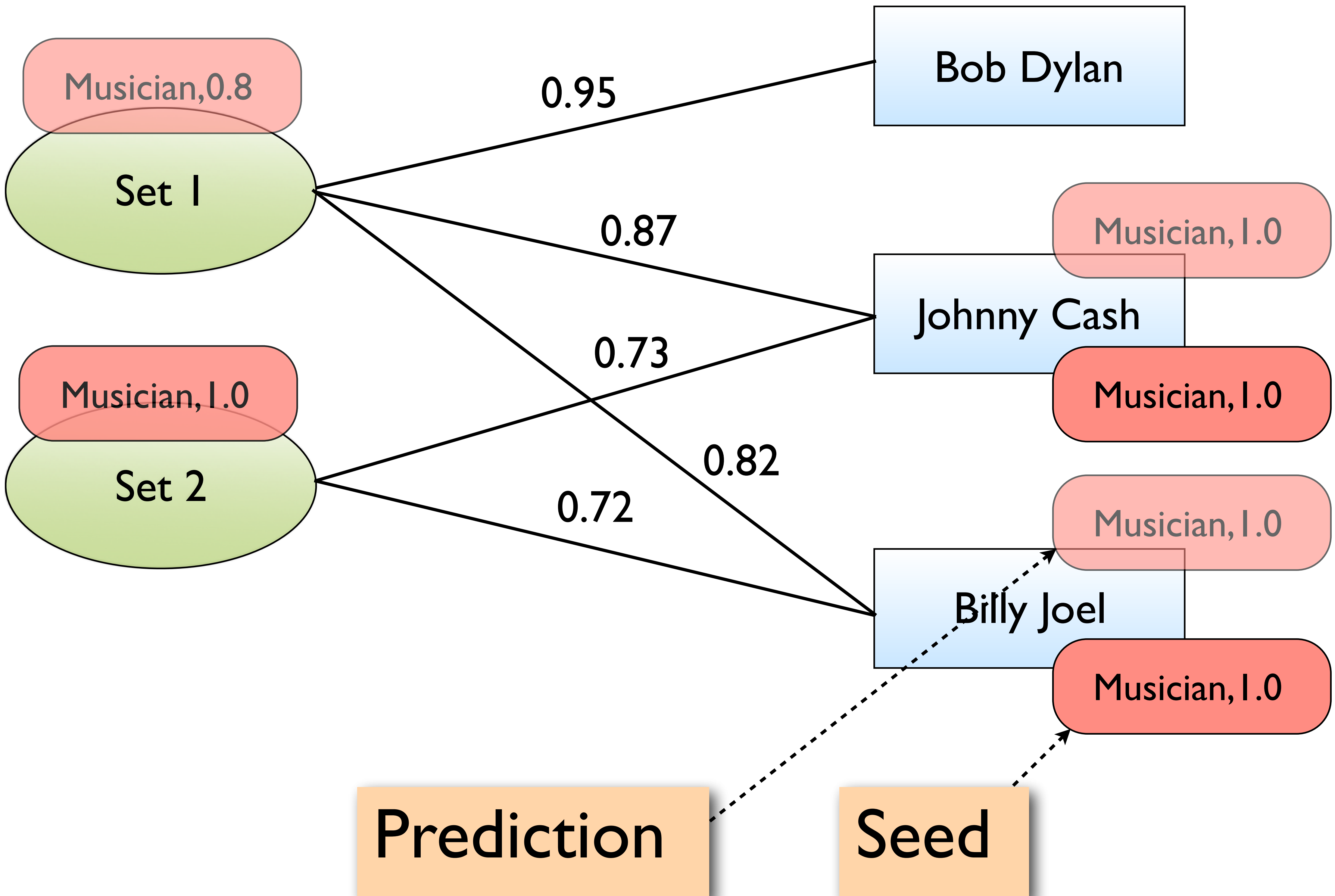
Graph Propagation



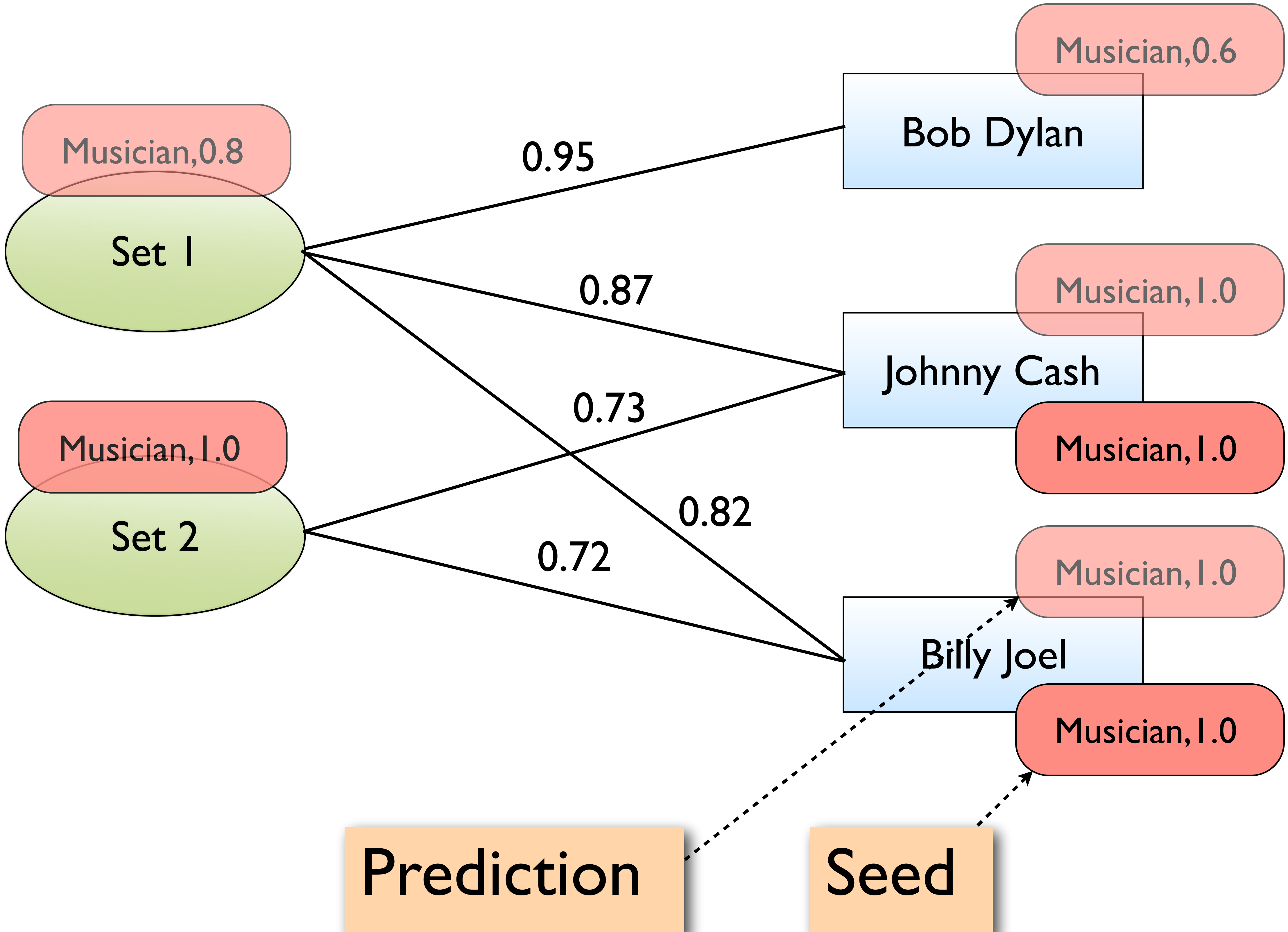
Graph Propagation



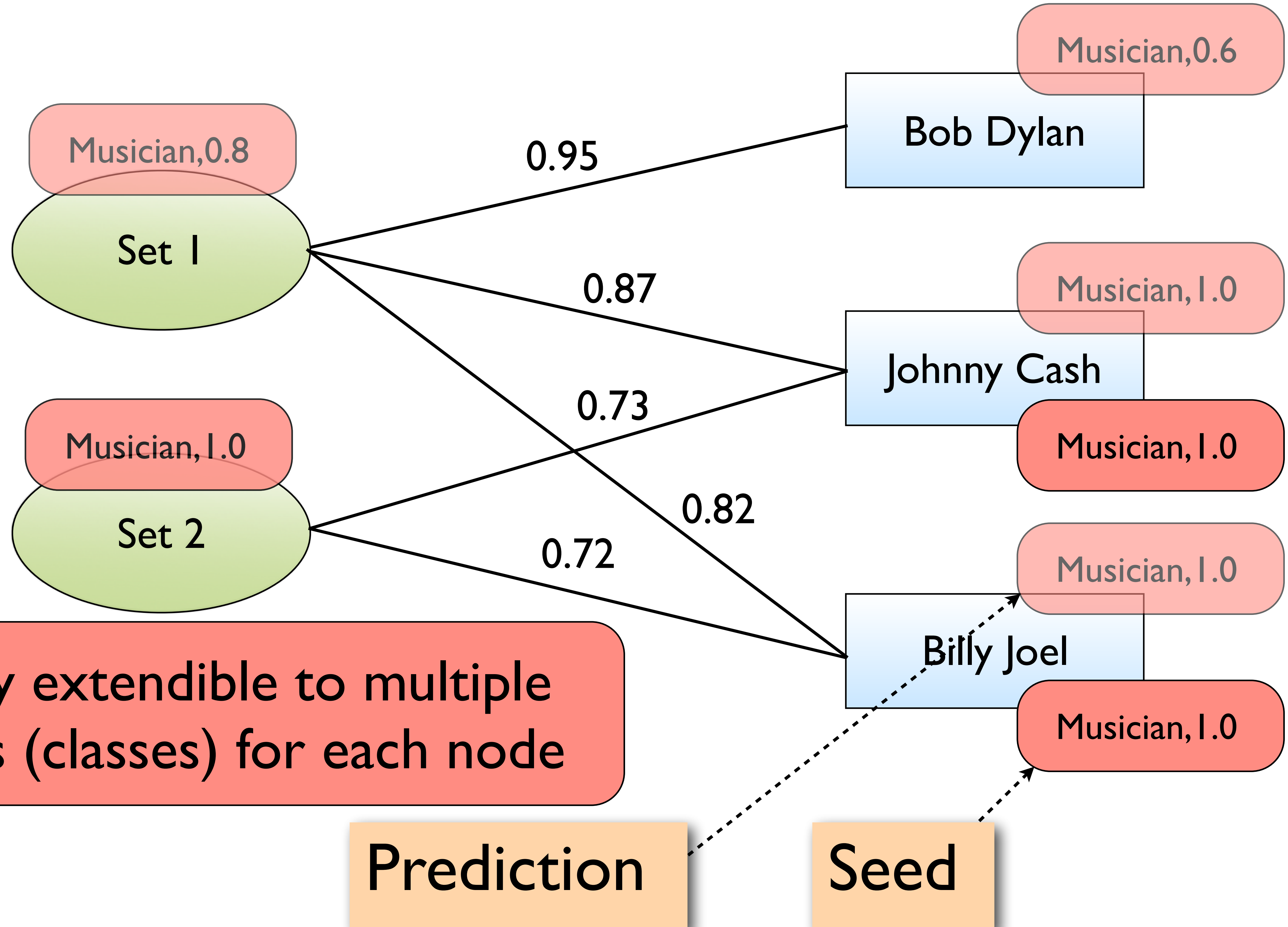
Graph Propagation



Graph Propagation



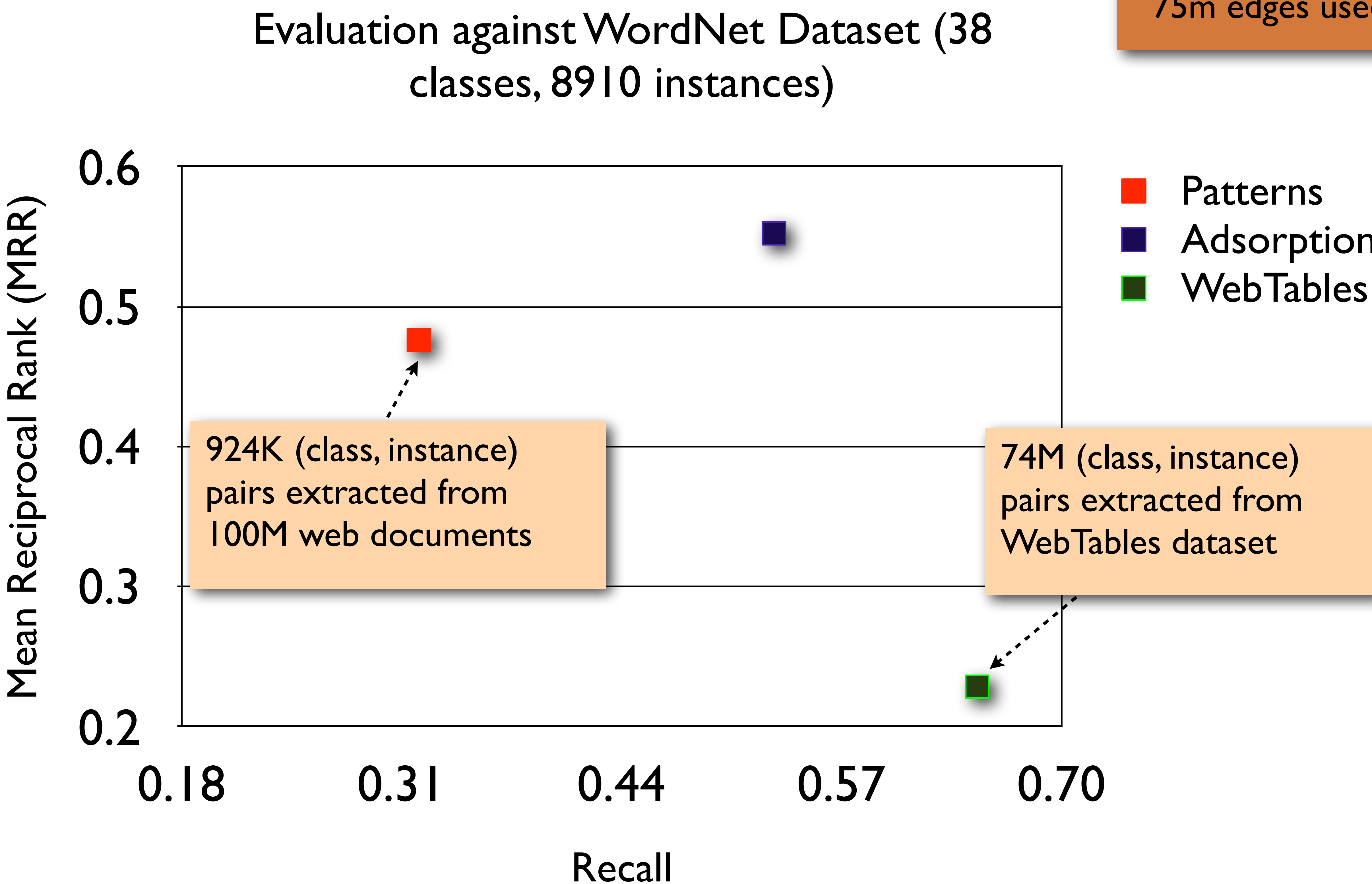
Graph Propagation



Easily extendible to multiple seeds (classes) for each node

Extraction for Known Instances

Graph with 1.4m nodes, 75m edges used.



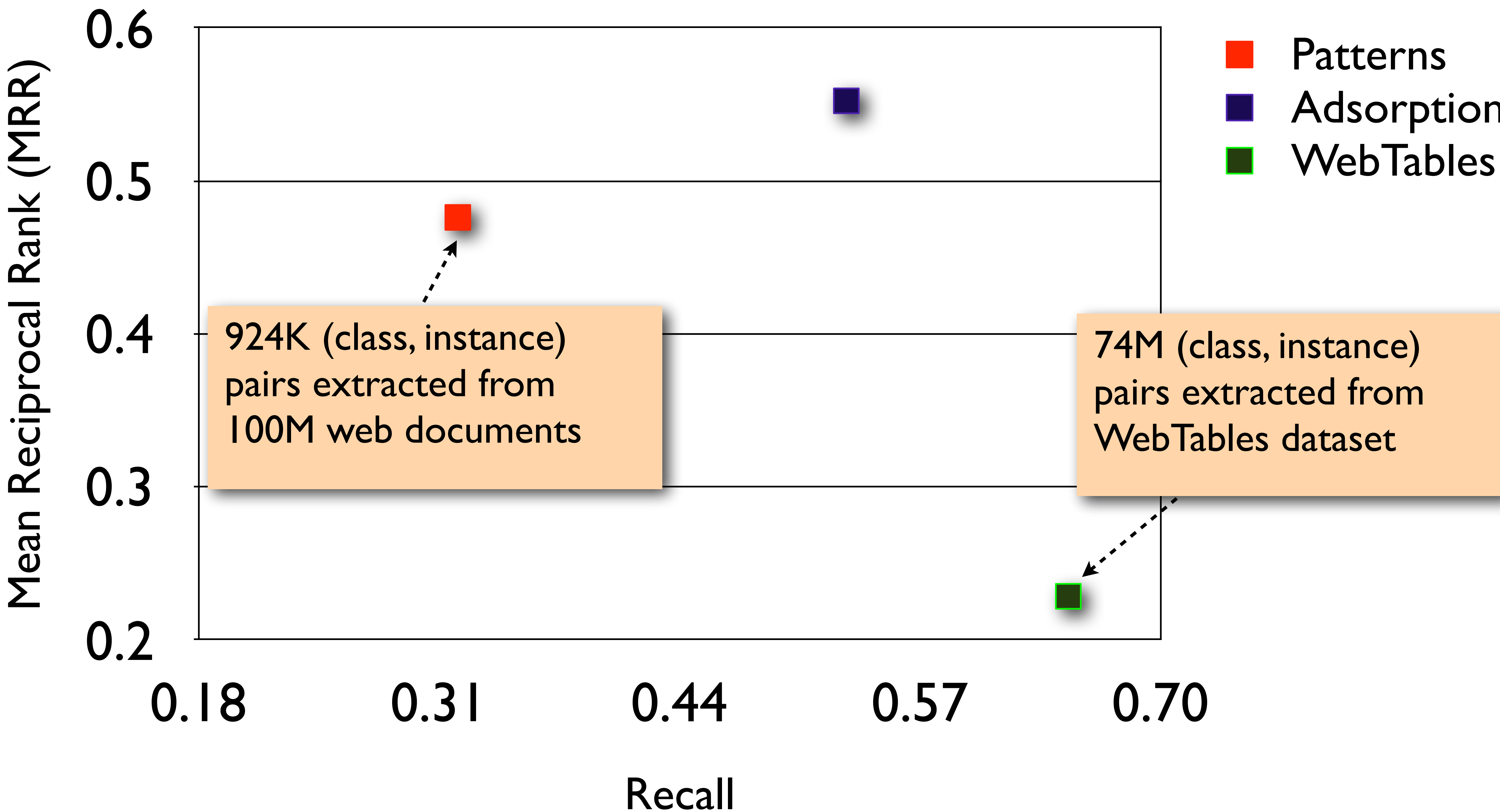
Extraction for Known Instances

Adsorption is able to assign ***better*** class labels to ***more*** instances.

Graph with 1.4m nodes, 75m edges used.

8

classes, 8910 instances)

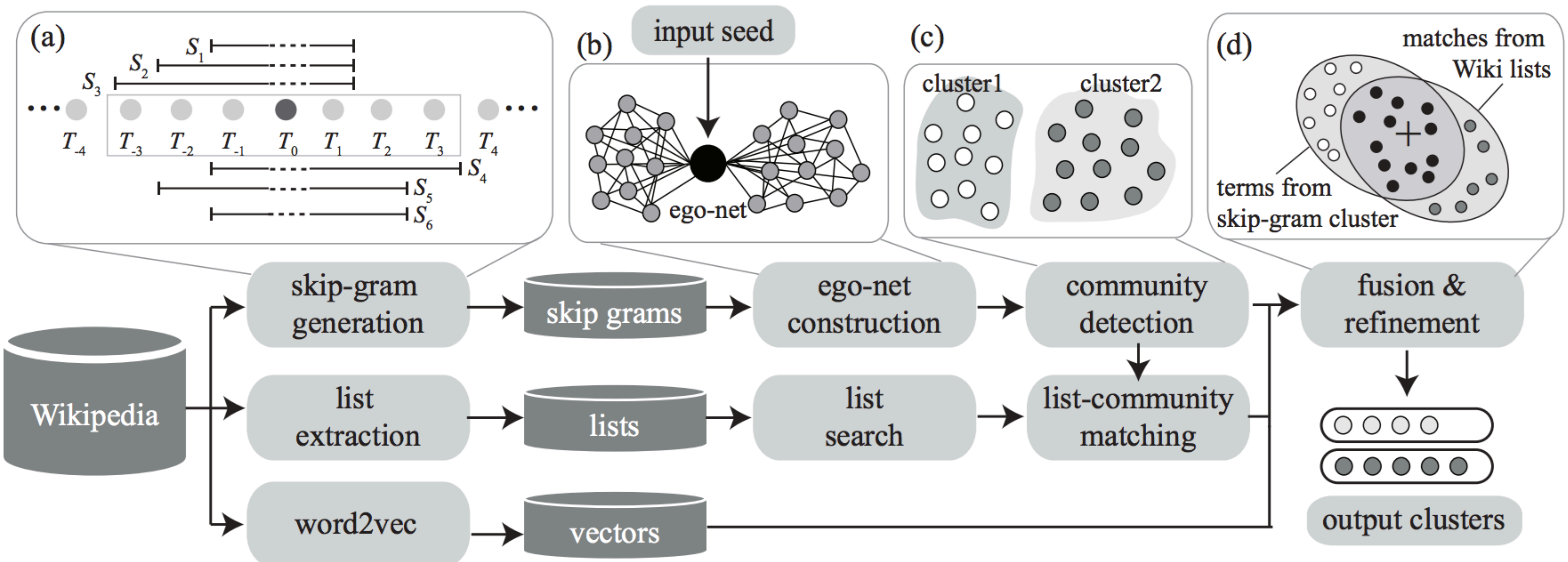


Extracted Pairs

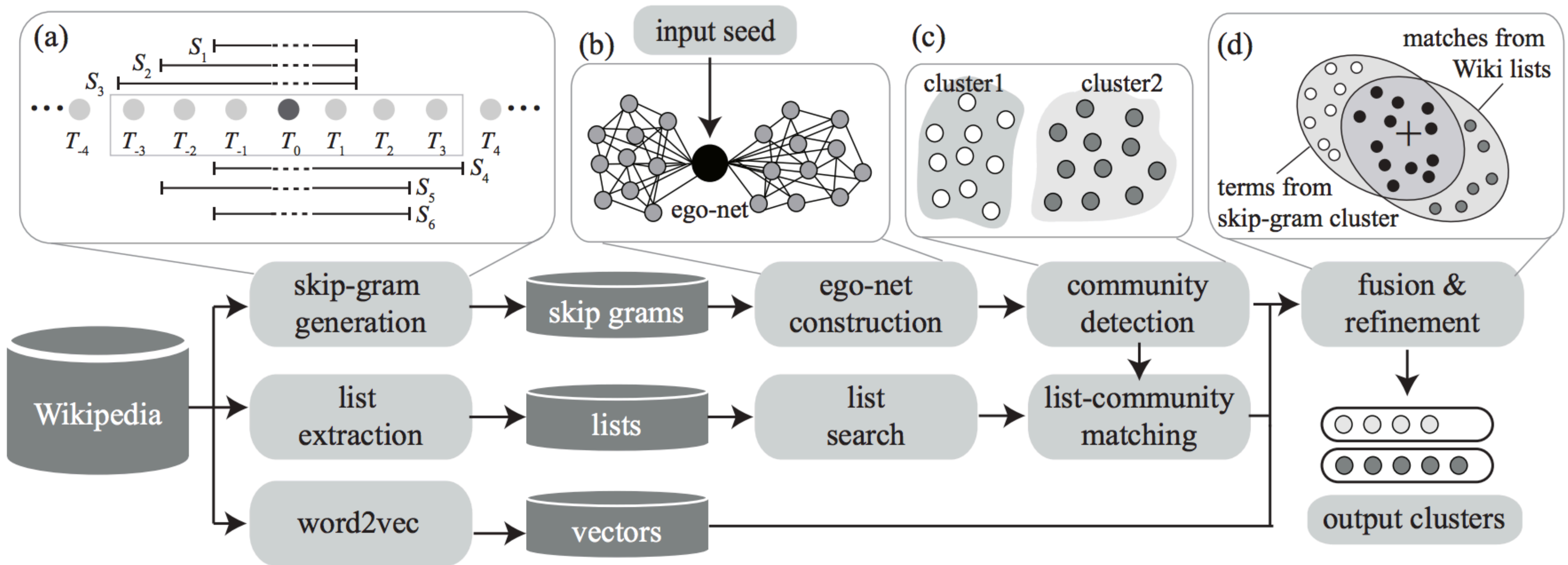
Total classes: **9081**

Class	Some non-seed Instances found
Scientific Journals	Journal of Physics, Nature, Structural and Molecular Biology, Sciences Sociales et sante, Kidney and Blood Pressure Research, American Journal of Physiology-Cell Physiology, …
NFL Players	Tony Gonzales, Thabiti Davis, Taylor Stubblefield, Ron Dixon, Rodney Hannan, …
Book Publishers	Small Night Shade Books, House of Ansari Press, Highwater Books, Distributed Art Publishers, Cooper Canyon Press, …

EgoSet [Rong et al., WSDM 2016]



EgoSet [Rong et al., WSDM 2016]



		1 seed			2 seeds			3 seeds			4 seeds		
		p@5	p@10	p@20	p@5	p@10	p@20	p@5	p@10	p@20	p@5	p@10	p@20
baseline	**SEAL**	-	-	-	0.208	0.169	0.138	0.368	0.312	0.269	0.393	0.342	0.298
	NeedleSeek	0.432	0.372	0.325	-	-	-	-	-	-	-	-	-
single	**WikiList**	0.369	0.331	0.292	0.313	0.295	0.250	0.401	0.340	0.284	0.379	0.366	0.325
	word2vec	0.360	0.296	0.249	0.317	0.271	0.219	0.389	0.313	0.247	0.431	0.373	0.320
fusion	**EgoSet-SG & WikiList**	0.465	0.413	0.358	0.357	0.316	0.272	0.366	0.325	0.280	0.447	0.374	0.329
	word2vec & WikiList	0.390	0.331	0.289	0.334	0.313	0.222	0.373	0.303	0.240	0.352	0.333	0.308
	EgoSet-ALL	**0.490**	**0.427**	**0.372**	**0.369**	**0.323**	**0.274**	**0.432**	**0.370**	**0.313**	**0.453**	**0.399**	**0.356**

Table 4: End-to-end performance evaluation.

Outline

13:00-13:15 Overview and motivation

13:15-13:45 Case study: NELL

13:45-14:00 Bootstrapped Entity Extraction

14:00-15:00 Open Relation Extraction & Canonicalization

15:00-15:30 **Coffee Break**

15:30-16:15 Distantly-supervised Relation Extraction

16:15-16:45 Knowledge Graph Embeddings

16:45-17:00 Conclusion & QA

Many OpenIE slides from Mausam

Two Types of Knowledge Graphs

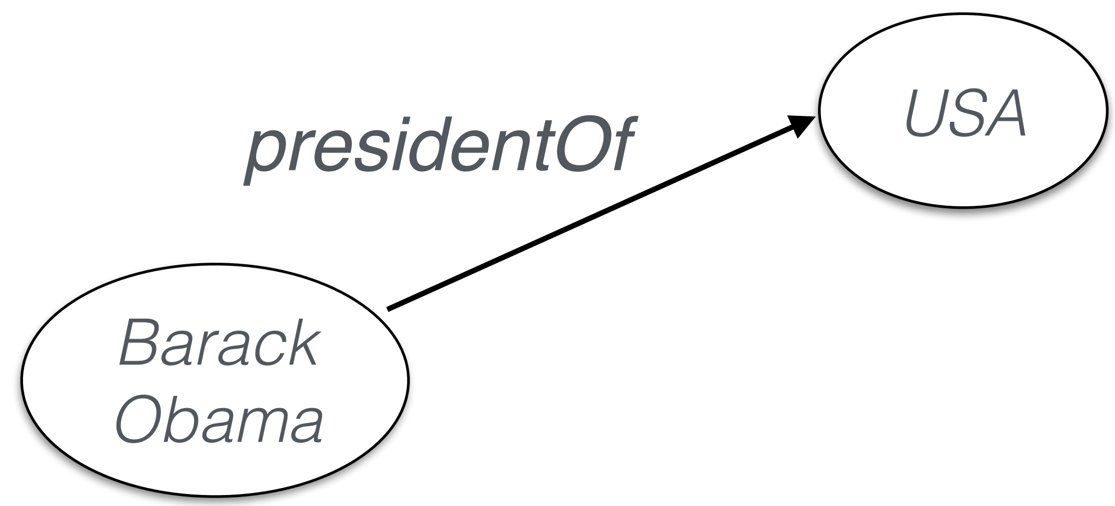
Two Types of Knowledge Graphs

“Obama was the President of USA.”

Two Types of Knowledge Graphs

"Obama was the President of USA."

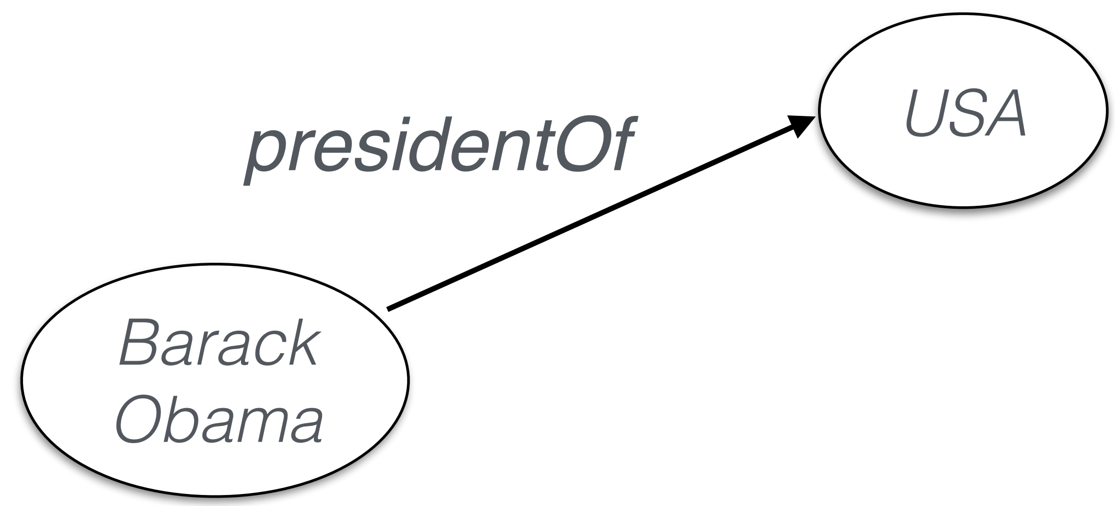
Ontological KG



Two Types of Knowledge Graphs

"Obama was the President of USA."

Ontological KG

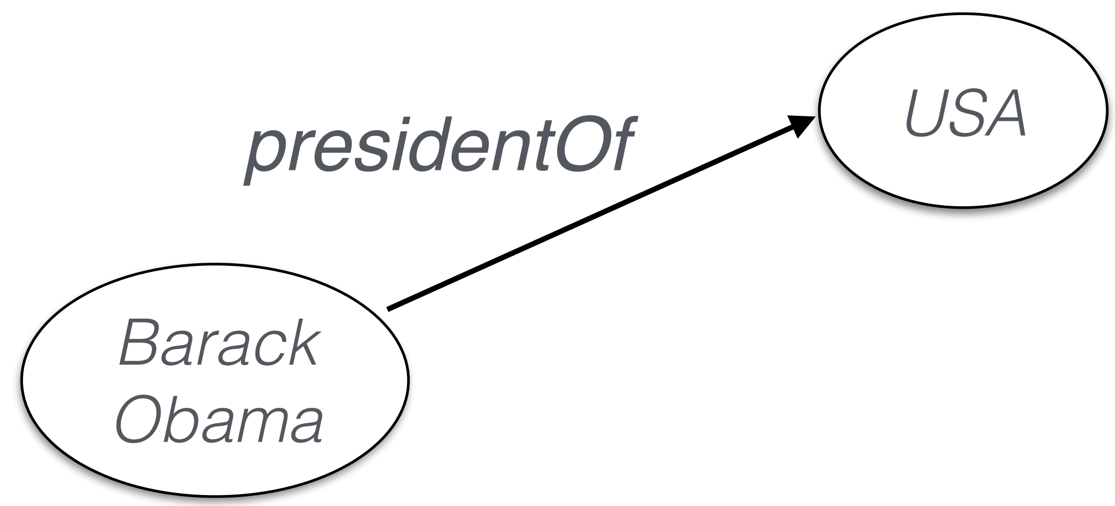


\+ high precision

Two Types of Knowledge Graphs

"Obama was the President of USA."

Ontological KG

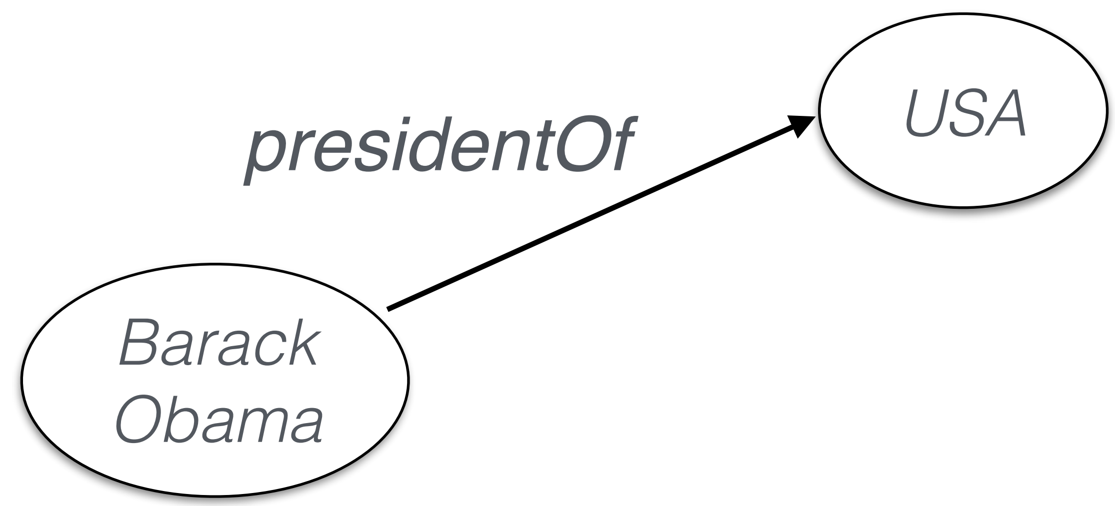


+ high precision
+ canonicalized/normalized

Two Types of Knowledge Graphs

"Obama was the President of USA."

Ontological KG

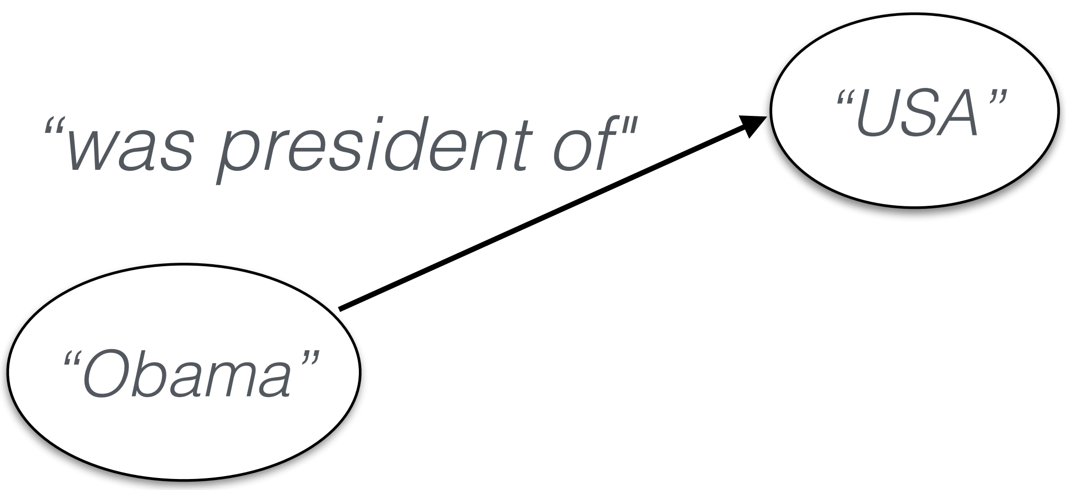


+ high precision
+ canonicalized/normalized
- requires supervision

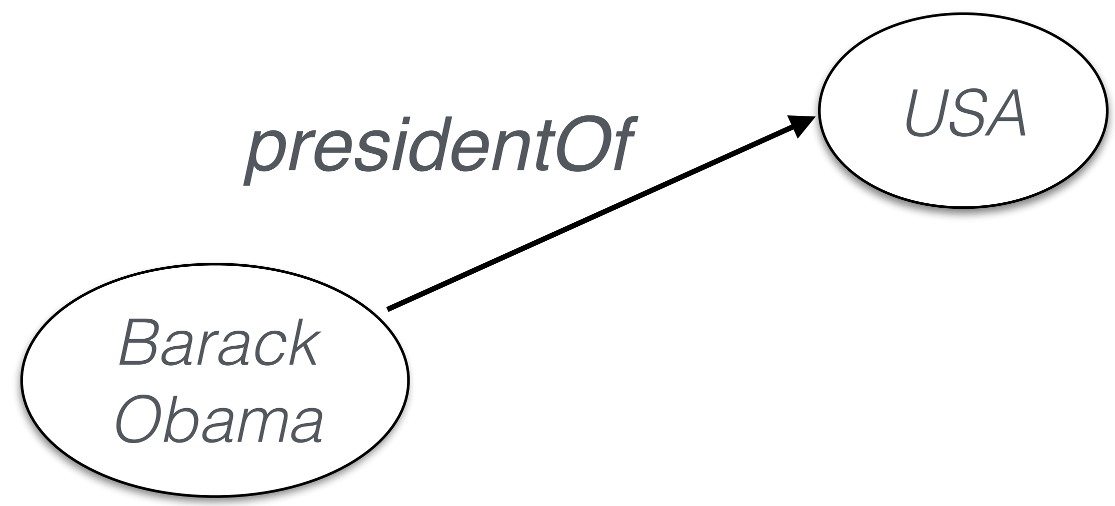
Two Types of Knowledge Graphs

"Obama was the President of USA."

Open KG
(Ontology Free)



Ontological KG

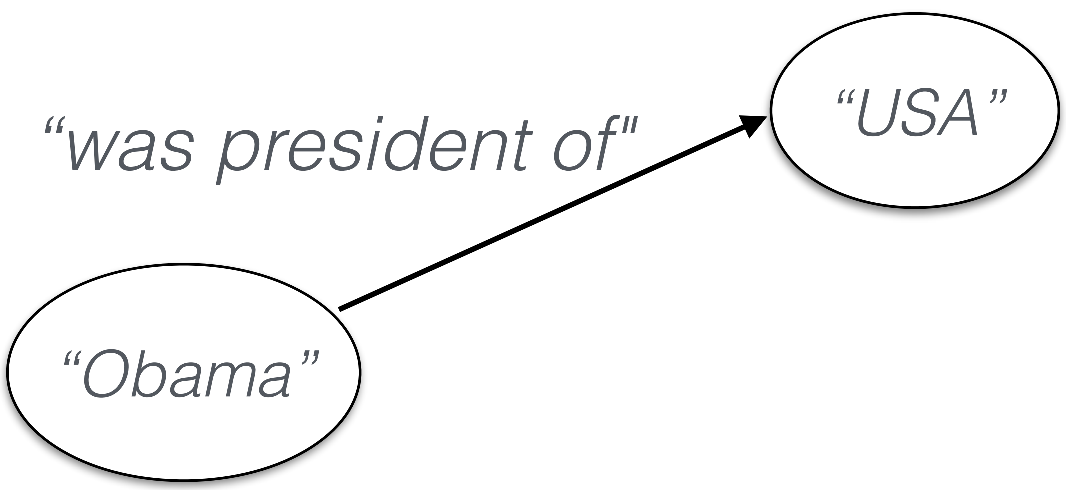


+ high precision
+ canonicalized/normalized
- requires supervision

Two Types of Knowledge Graphs

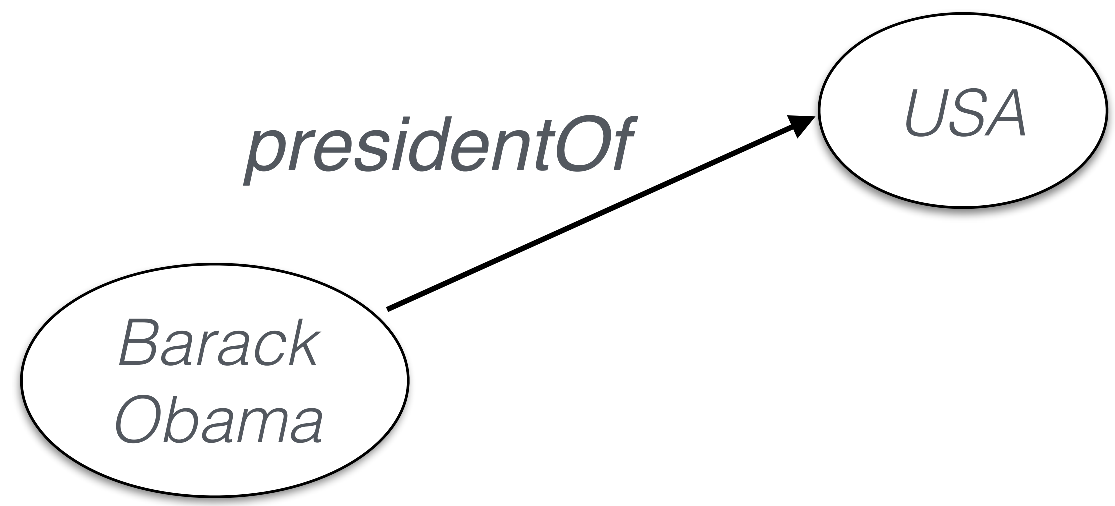
"Obama was the President of USA."

Open KG
(Ontology Free)



+ easy to build, available tools

Ontological KG

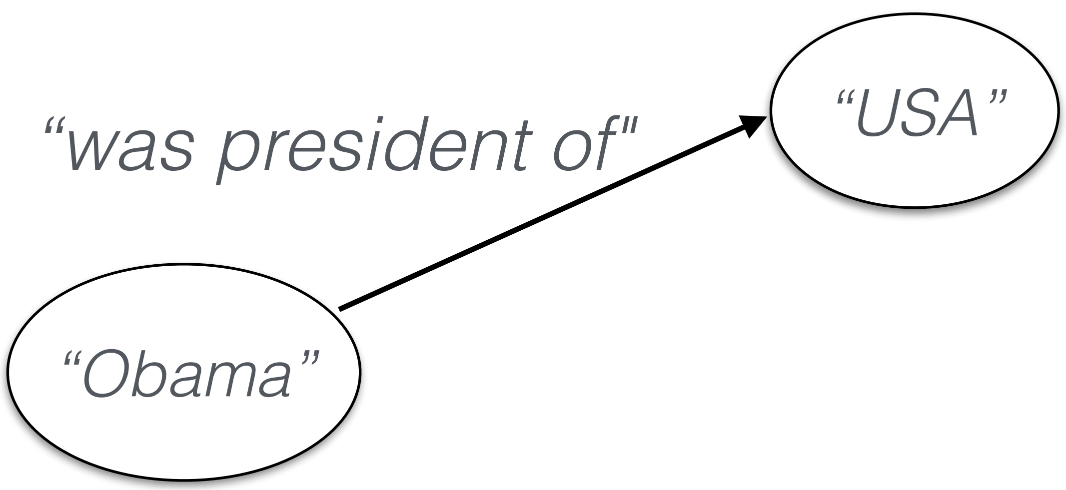


+ high precision
+ canonicalized/normalized
- requires supervision

Two Types of Knowledge Graphs

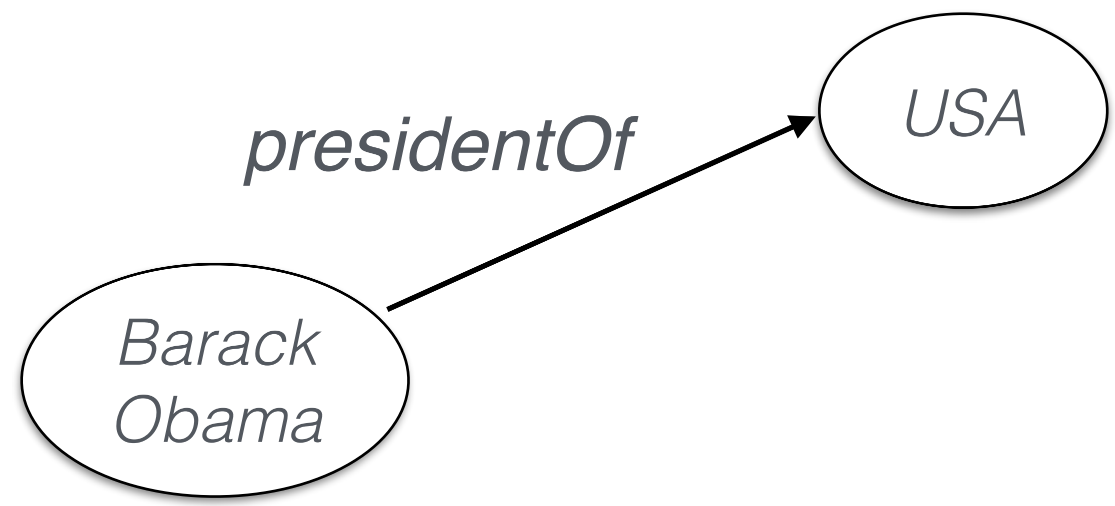
"Obama was the President of USA."

Open KG
(Ontology Free)



- \+ easy to build, available tools
- \+ high recall

Ontological KG

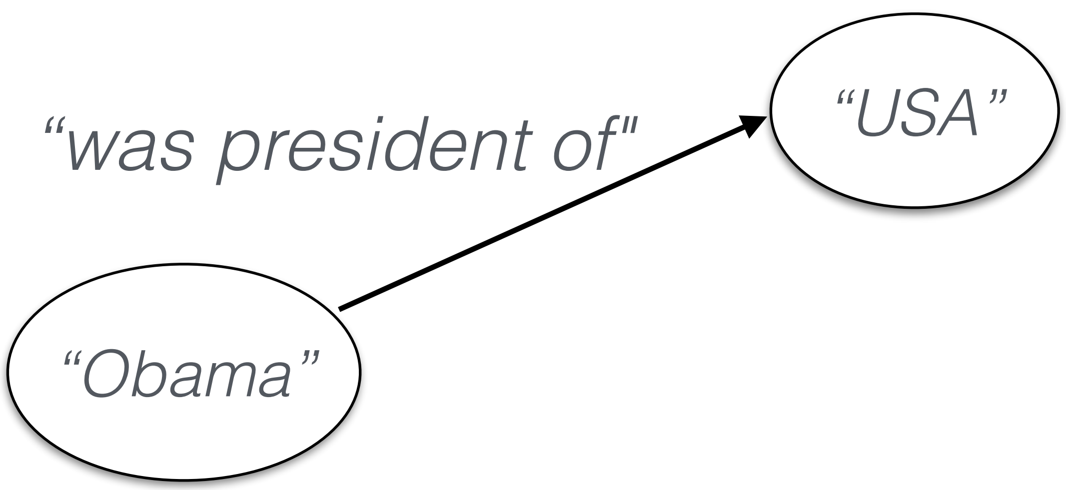


+ high precision
+ canonicalized/normalized
- requires supervision

Two Types of Knowledge Graphs

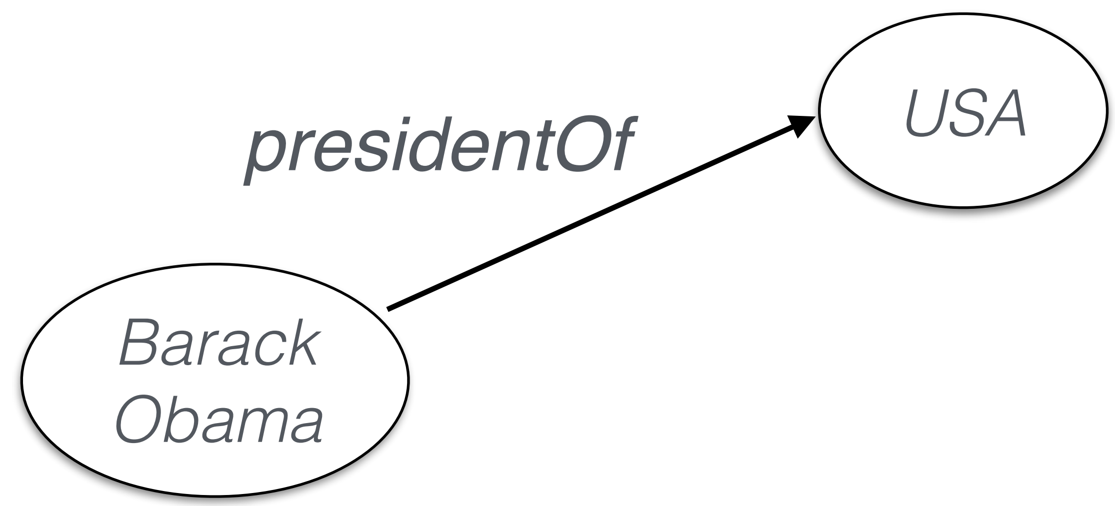
“Obama was the President of USA.”

Open KG
(Ontology Free)



- + easy to build, available tools
- + high recall
- - fragmented (more later)

Ontological KG



+ high precision
+ canonicalized/normalized
- requires supervision

Machine Reading at Web Scale

- **A “universal schema” is impossible**

Machine Reading at Web Scale

- **A "universal schema" is impossible**
- **Global consistency is like world peace**

Machine Reading at Web Scale

- **A "universal schema" is impossible**
- **Global consistency is like world peace**

- **Ontological "glass ceiling"**
  - Limited vocabulary
  - Pre-determined predicates
  - Swamped by reading at scale!

Motivation

- General purpose
  - hundreds of thousands of relations
  - thousands of domains
- Scalable: computationally efficient
  - huge body of text on Web and elsewhere
- Scalable: minimal manual effort
  - large-scale human input impractical
- Knowledge needs not anticipated in advance
  - rapidly retargetable

Motivation

- General purpose
  - hundreds of thousands of relations
  - thousands of domains
- Scalable: computationally efficient
  - huge body of text on Web and elsewhere
- Scalable: minimal manual effort
  - large-scale human input impractical
- Knowledge needs not anticipated in advance
  - rapidly retargetable

Open IE Guiding Principles

- Domain independence
  - Training for each domain/fact type not feasible
- Scalability
  - Ability to process large number of documents fast
- Coherence
  - Readability important for human interactions

Open Information Extraction

Extracting information from natural language text

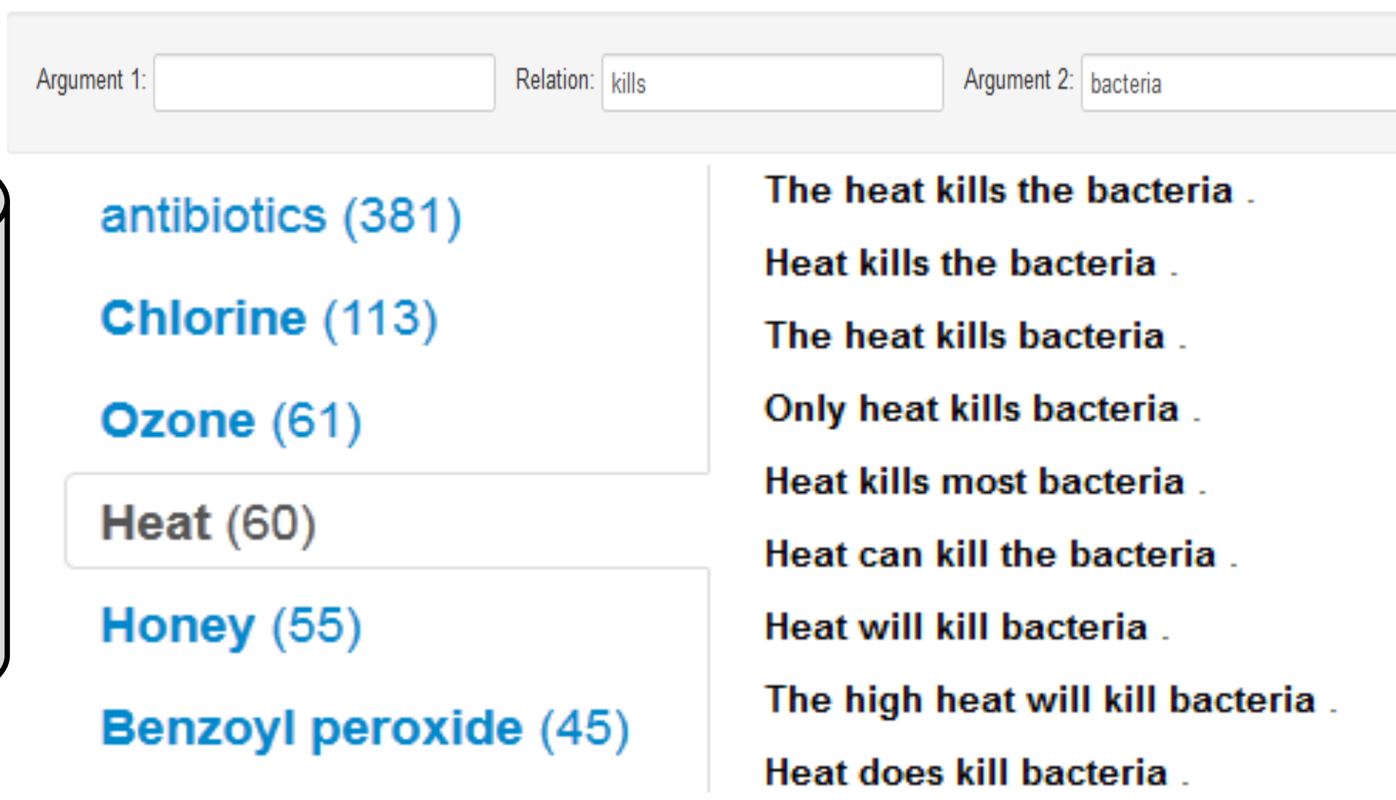
for *all* relations in *all* domains in a *few* passes.

"When Saddam Hussain invaded Kuwait in 1990, the international.."

↓ Open IE

(Saddam Hussain, invaded, Kuwait)

(Google, acquired, Youtube)
(Oranges, contain, Vitamin C)
(Edison, invented, phonograph)
…



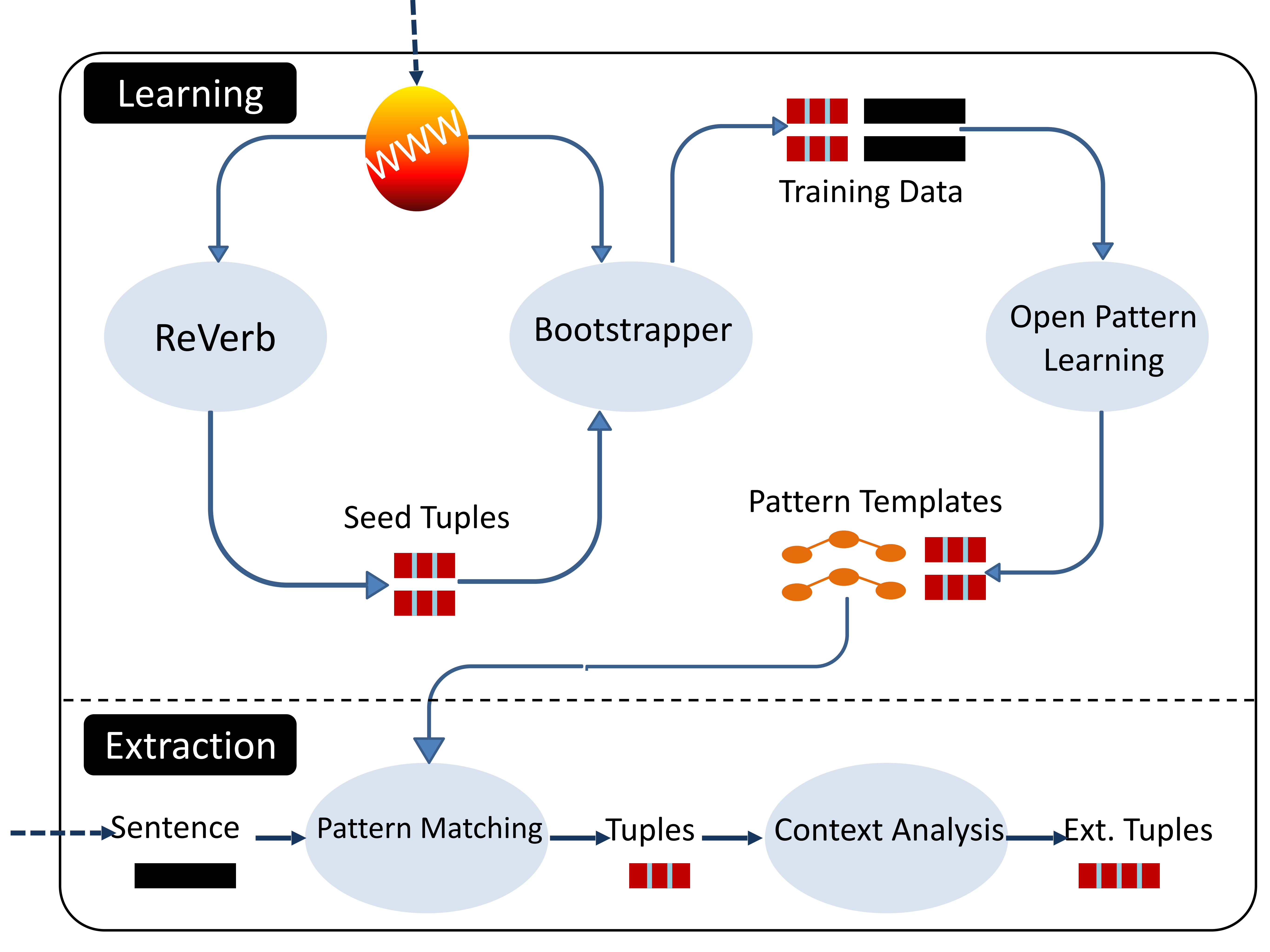
Open vs. Traditional IE

	Traditional IE	Open IE
Input:	Corpus + Hand-labeled Data	Corpus + Existing resources
Relations:	Specified in Advance	Discovered Automatically
Complexity:	O(*D* * ***R***) *R* relations	O(*D*) *D* documents
Output:	relation-specific	**Relation-independent**

Open Information Extraction

- 2007: Textrunner (~Open IE 1.0)
  - CRF and self-training
- 2010: ReVerb (~Open IE 2.0)
  - POS-based relation pattern
- 2012: OLLIE (~Open IE 3.0)
  - Dep-parse based extraction; nouns; attribution
- 2014: Open IE 4.0
  - SRL-based extraction; temporal, spatial…
- 2016 [@IITD]: Open IE 5.0
  - compound noun phrases, numbers, lists

increasing
precision,
recall,
expressiveness


Learning
www
Training Data
ReVerb
Bootstrapper
Open Pattern
Learning
Seed Tuples
Pattern Templates
Extraction
Sentence
Pattern Matching
Tuples
Context Analysis
Ext. Tuples

Context Analysis

"John refused to visit Vegas."

(John, refused to visit, Vegas)

"Early astronomers believed that the earth is the center of the universe."

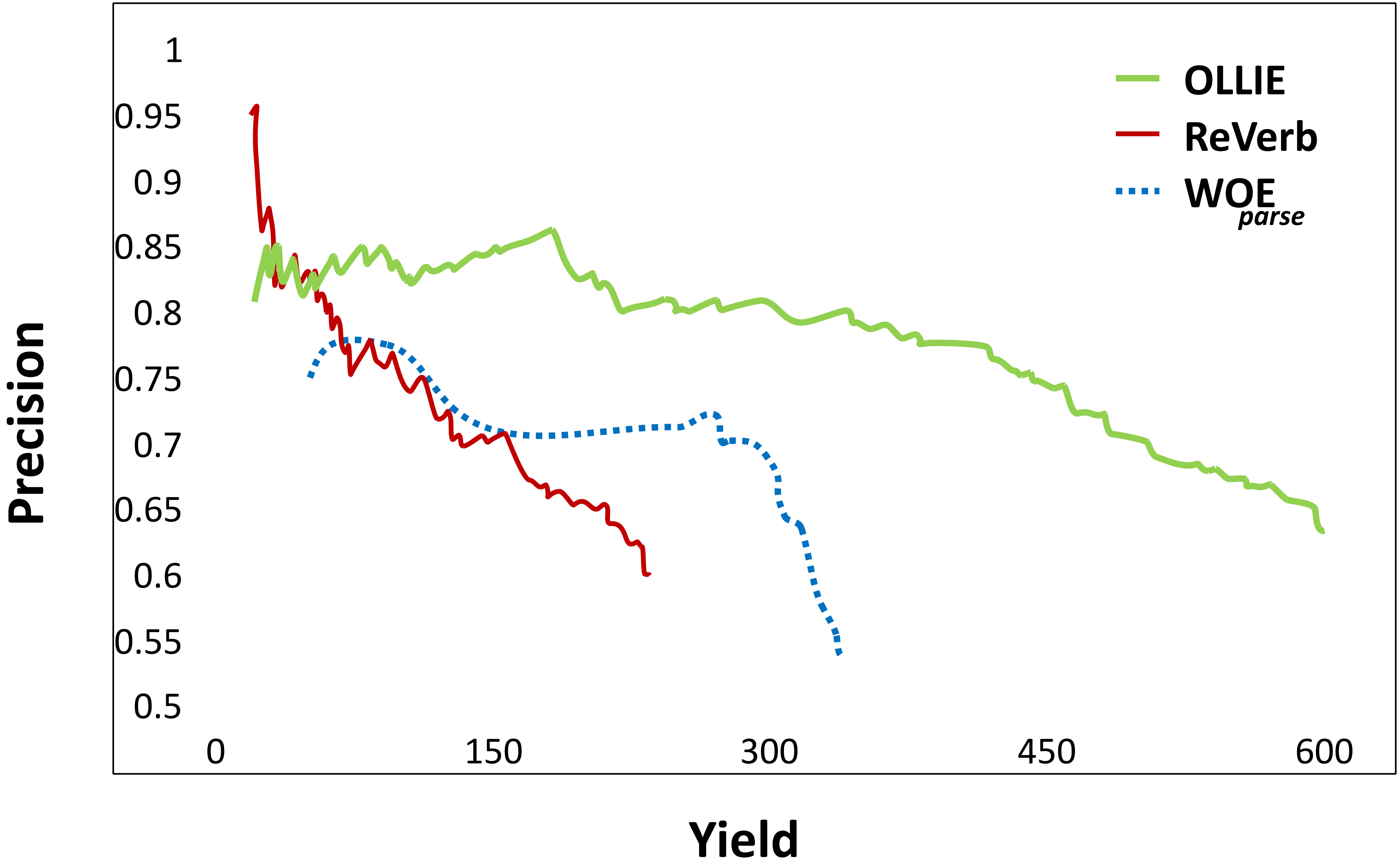
[(earth, is the center of, universe) Attribution: early astronomers]

"If she wins California, Hillary will be the nominated presidential candidate."

[(Hillary, will be nominated, presidential candidate) Modifier: if she wins California]

Evaluation

[Mausam, Schmitz, Bart, Soderland, Etzioni - EMNLP'12]



Take Homes

- Bootstrapping based on ReVerb
  - Look for args as well as relations when bootstrapping
- Generalization
  - Syntactic and semantic generalizations of learned patterns
- Context around an extraction
  - Obtains superior precision than ReVerb
- Syntactically different ways of expressing a relation
  - Obtains much higher recall than ReVerb

Numerical Open IE

[Saha, Pal, Mausam ACL'17]

"Venezuela with its inflation rate 96% is suffering from a major…"

↓ Numerical Open IE

(Venezuela, inflation rate, 96 %)

"Grand Trunk Road is 1,005 kms long."

↓ Numerical Open IE

(Grand Trunk Road, has length, 1005 kms)

OpenIE v5:
https://github.com/dair-iitd/OpenIE-standalone

Open KG Canonicalization

Open KGs

Open KGs

- Work in an “ontology free” setting.

Open KGs

- Work in an “ontology free” setting.
- Extract **<noun-phrase, relation-phrase, noun-phrase>** triples from each sentence
  - *Obama was the President of US. => (Obama, was president of, USA)*

Open KGs

- Work in an “ontology free” setting.
- Extract **<noun-phrase, relation-phrase, noun-phrase>** triples from each sentence
  - *Obama was the President of US. => (Obama, was president of, USA)*
- Examples: TextRunner, ReVerb, Ollie, etc.

Open KGs

- Work in an “ontology free” setting.
- Extract **<noun-phrase, relation-phrase, noun-phrase>** triples from each sentence
    - *Obama was the President of US. => (Obama, was president of, USA)*
- Examples: TextRunner, ReVerb, Ollie, etc.
- Issues:
    - May store redundant and split facts

        <Bangalore, capital-of, Karnataka>

        <Bengaluru, has-population, 11 million>

        <Mysore, city-in, Karnataka>

Open KGs

- Work in an “ontology free” setting.
- Extract **<noun-phrase, relation-phrase, noun-phrase>** triples from each sentence
  - *Obama was the President of US. => (Obama, was president of, USA)*
- Examples: TextRunner, ReVerb, Ollie, etc.
- Issues:
  - May store redundant and split facts

    <Bangalore, capital-of, Karnataka>

    <Bengaluru, has-population, 11 million>

    <Mysore, city-in, Karnataka>
- Need to canonicalize Open KGs

NP Canonicalization

NP Canonicalization

Barack Obama, Mr. Obama, George Bush, Mumbai, Bombay, Madrid

NP Canonicalization

Barack Obama, Mr. Obama, George Bush, Mumbai, Bombay, Madrid

Barack Obama
Mr. Obama

George Bush

Madrid

Mumbai
Bombay

[Galarraga et al., 2014]

- Canonicalize Open KG by clustering synonymous nouns phrases.
- Uses several types of measures for defining similarity between synonymous noun phrases
- After noun phrase canonicalization, AMIE [Galarraga et al., 2013] is employed for canonicalizing relations

▸ **IDF Token Overlap:**

$$f(m, m') = \frac{\sum_{x \in w(m) \cap w(m')} \log\left(1 + df(x)\right)^{-1}}{\sum_{x \in w(m) \cup w(m')} \log\left(1 + df(x)\right)^{-1}}$$

CESI [Vashishth et al., 2018]

CESI [Vashishth et al., 2018]

- Embeds noun and relation phrases followed by clustering for canonicalizing Open KGs

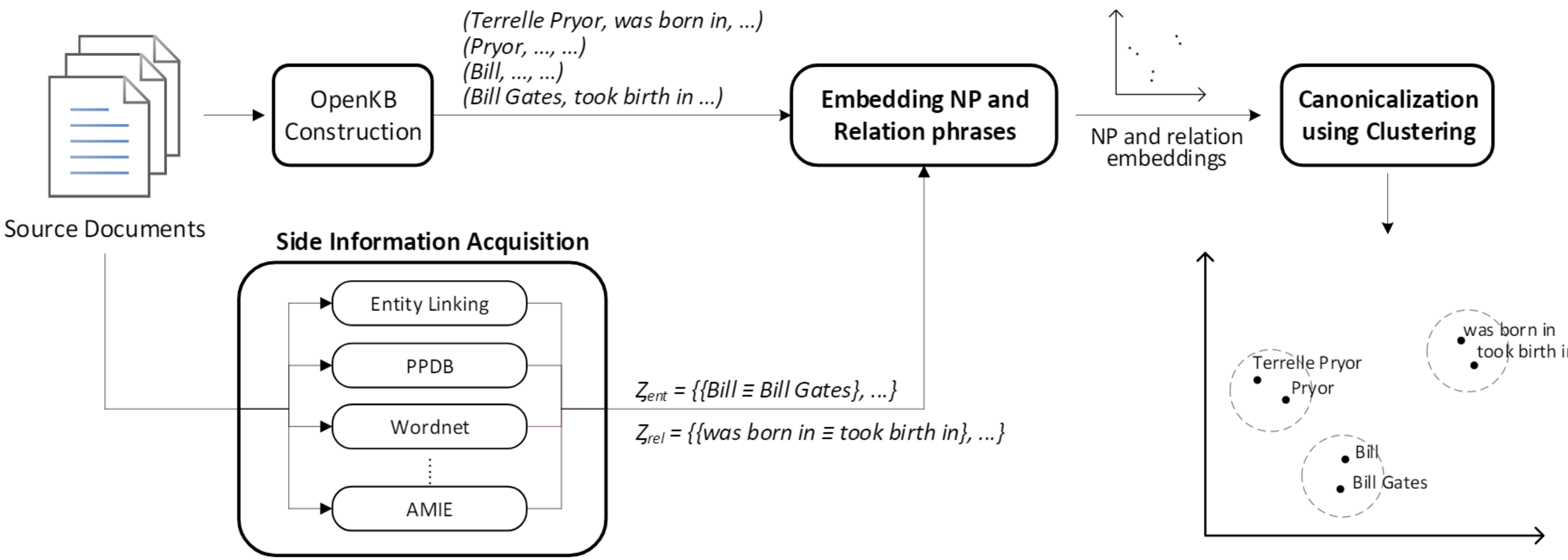
CESI [Vashishth et al., 2018]

- Embeds noun and relation phrases followed by clustering for canonicalizing Open KGs
- Jointly canonicalizes noun and relation phrases while utilizing relevant side information

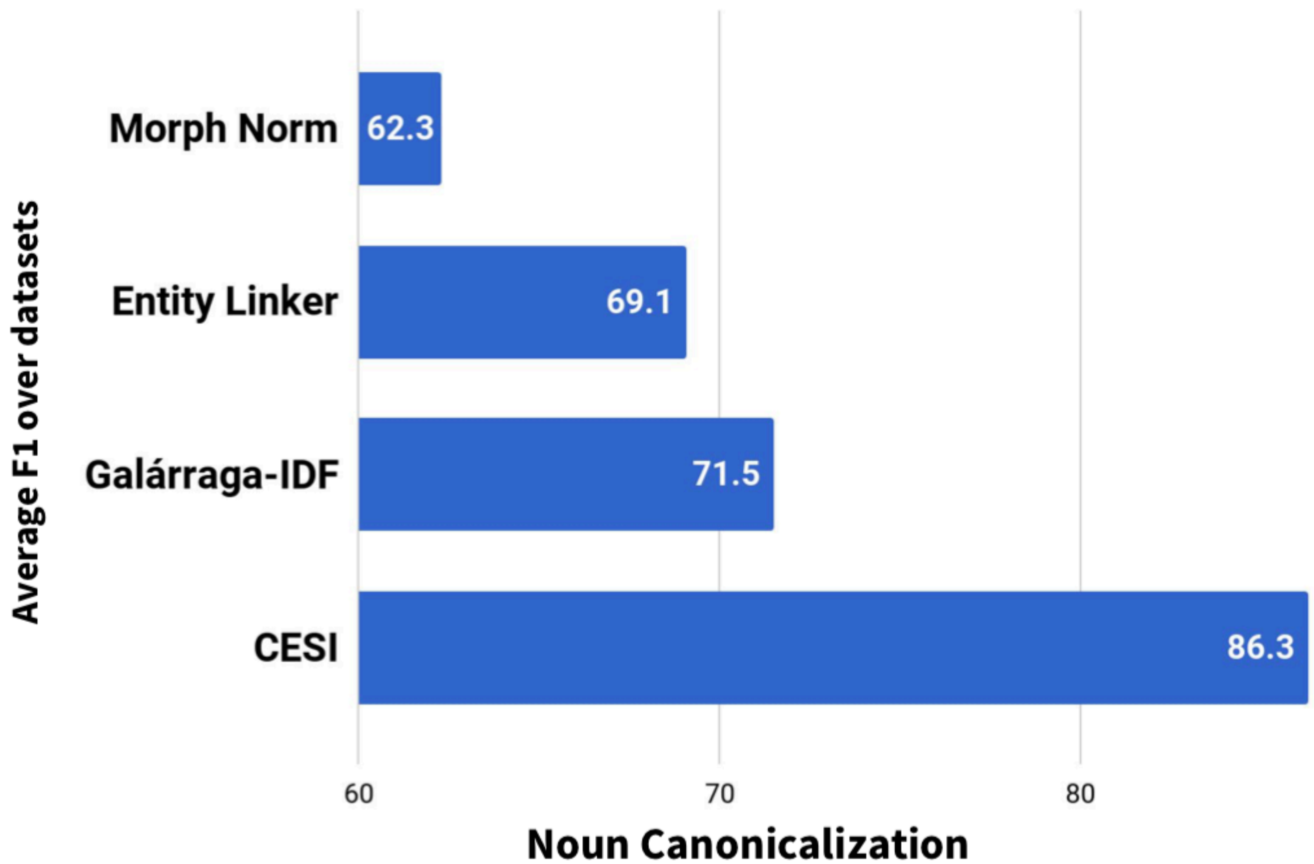
CESI [Vashishth et al., 2018]

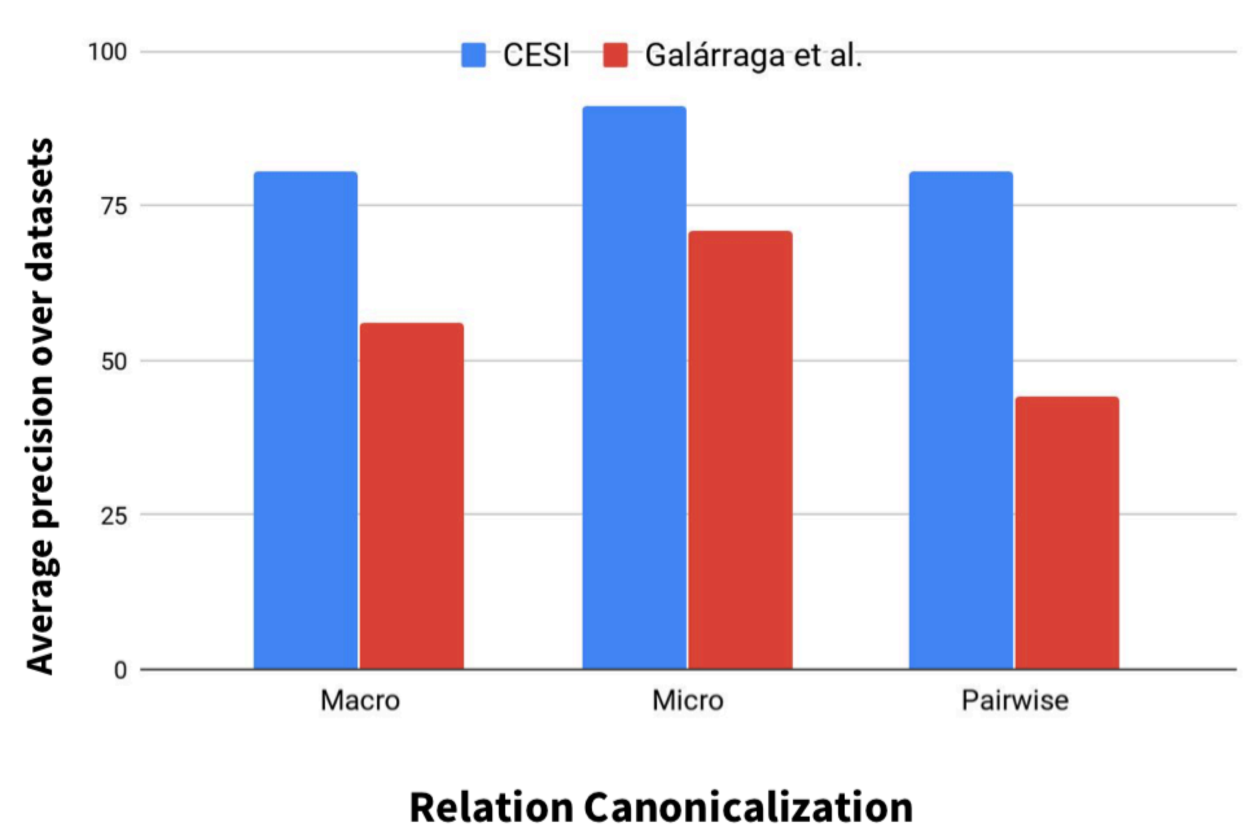
- Embeds noun and relation phrases followed by clustering for canonicalizing Open KGs
- Jointly canonicalizes noun and relation phrases while utilizing relevant side information
- **Side Information Acquisition**: Gathers various NP and relation phrase side information for each triple in KG
  - e.g., entity linking, paraphrasing, token overlap etc.

CESI [Vashishth et al., 2018]

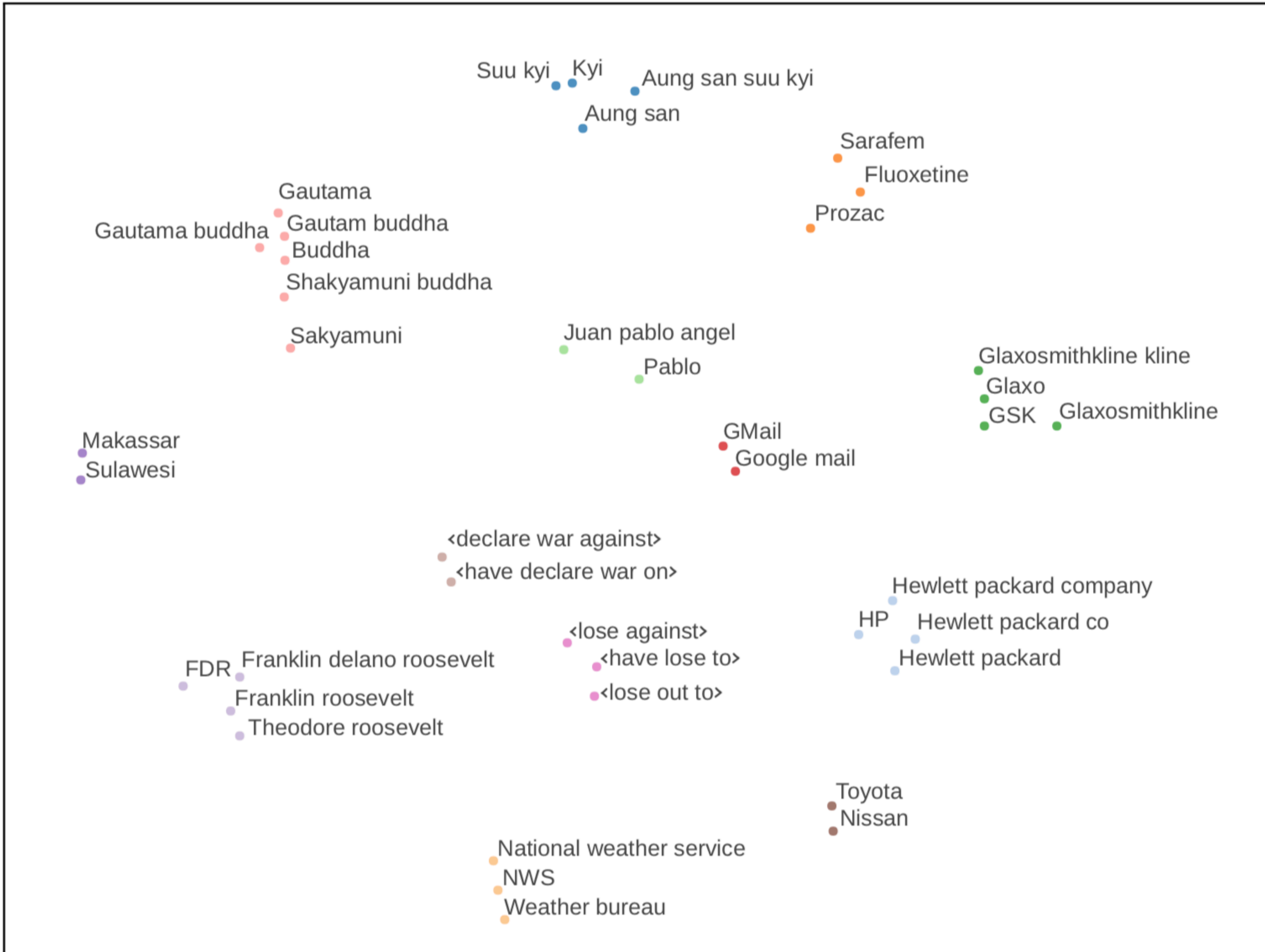


Canonicalization Results





CESI Code: https://github.com/malllabiisc/cesi


Suu kyi
Kyi
Aung san suu kyi
Aung san
Sarafem
Fluoxetine
Prozac
Gautama
Gautama buddha
Gautam buddha
Buddha
Shakyamuni buddha
Sakyamuni
Juan pablo angel
Pablo
Glaxosmithkline kline
Glaxo
GSK
Glaxosmithkline
GMail
Google mail
Makassar
Sulawesi
<declare war against>
<have declare war on>
Hewlett packard company
HP
Hewlett packard co
Hewlett packard
<lose against>
<have lose to>
<lose out to>
FDR
Franklin delano roosevelt
Franklin roosevelt
Theodore roosevelt
Toyota
Nissan
National weather service
NWS
Weather bureau

Relation Schema Induction

Domain-specific Knowledge Graphs (KG)

Domain-specific Knowledge Graphs (KG)

- Need KGs in specific domains (e.g., insurance, automotives, etc.)

Domain-specific Knowledge Graphs (KG)

- Need KGs in specific domains (e.g., insurance, automotives, etc.)

- General purpose KGs (e.g., Freebase, YAGO, NELL, etc.) are good starting points, but often not enough

Domain-specific Knowledge Graphs (KG)

- Need KGs in specific domains (e.g., insurance, automotives, etc.)

- General purpose KGs (e.g., Freebase, YAGO, NELL, etc.) are good starting points, but often not enough

- Problem: how to build KG out of documents from a given domain, with minimal supervision?

Relation Schema Induction

Relation Schema Induction

- Relation Schemas [e.g., *undergo(Patient, Surgery)*]
  - starting point in ontological KG construction
  - prepared by experts: expensive and incomplete

Relation Schema Induction

- Relation Schemas [e.g., *undergo(Patient, Surgery)*]
  - starting point in ontological KG construction
  - prepared by experts: expensive and incomplete

"… John underwent angioplasty last Tuesday …"
"… Sam will undergo Tonsillectomy …"
…
"… cells that undergo meiosis …"

…
undergo(Patient, Surgery)
undergo(Cell, Division)
…

Relation Schema Induction

- Relation Schemas [e.g., *undergo(Patient, Surgery)*]
  - starting point in ontological KG construction
  - prepared by experts: expensive and incomplete

"… John underwent angioplasty last Tuesday …"
"… Sam will undergo Tonsillectomy …"
…
"… cells that undergo meiosis …"

…
undergo(Patient, Surgery)
undergo(Cell, Division)
…

How to automatically identify relations and their schemas from domain documents?

KB-LDA [Movshovitz-Attias and Cohen, 2015]

- A topic modeling approach for KB schema induction
- Learns both latent hierarchical structure of categories and latent semantic relations between categories

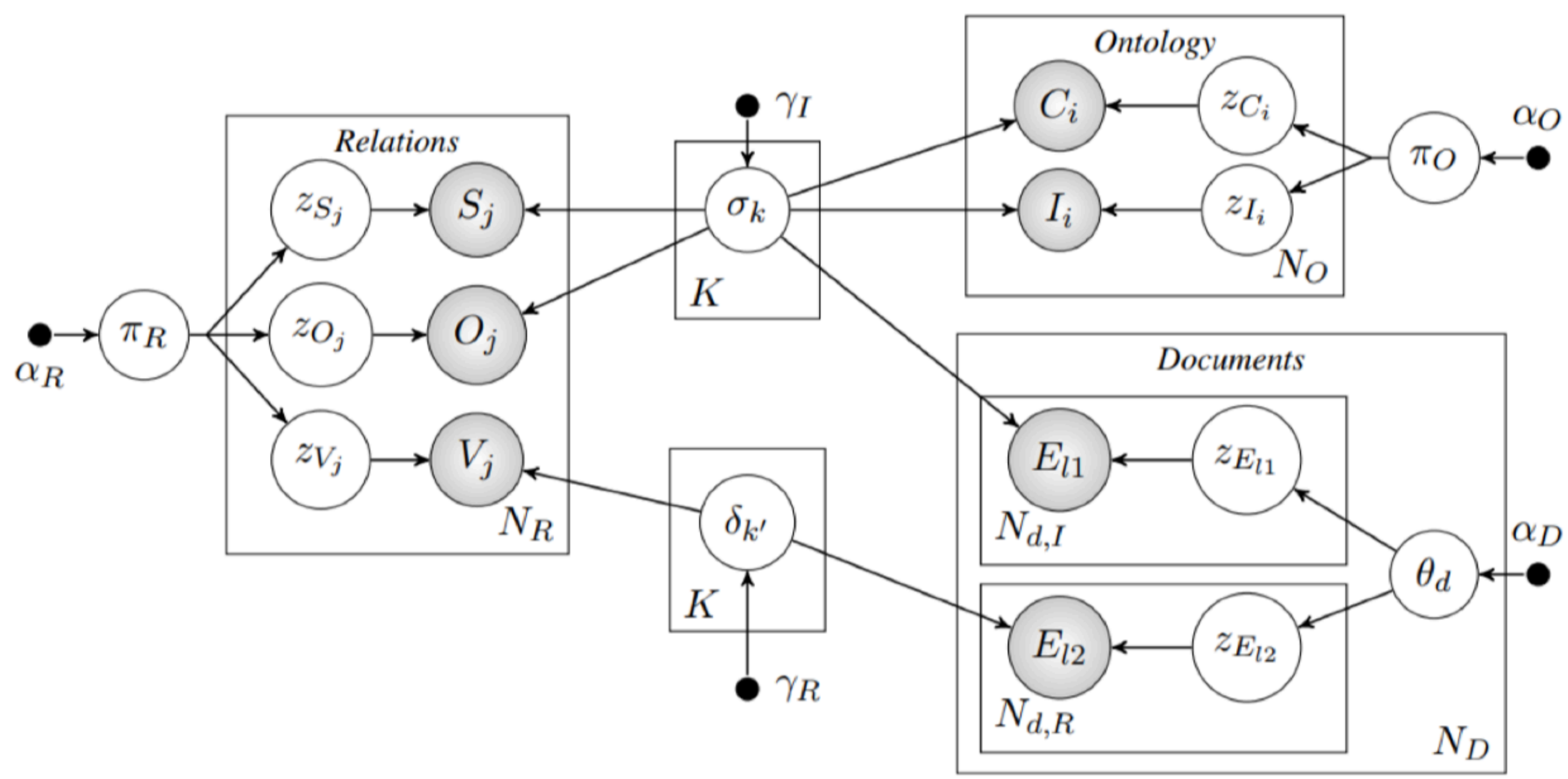
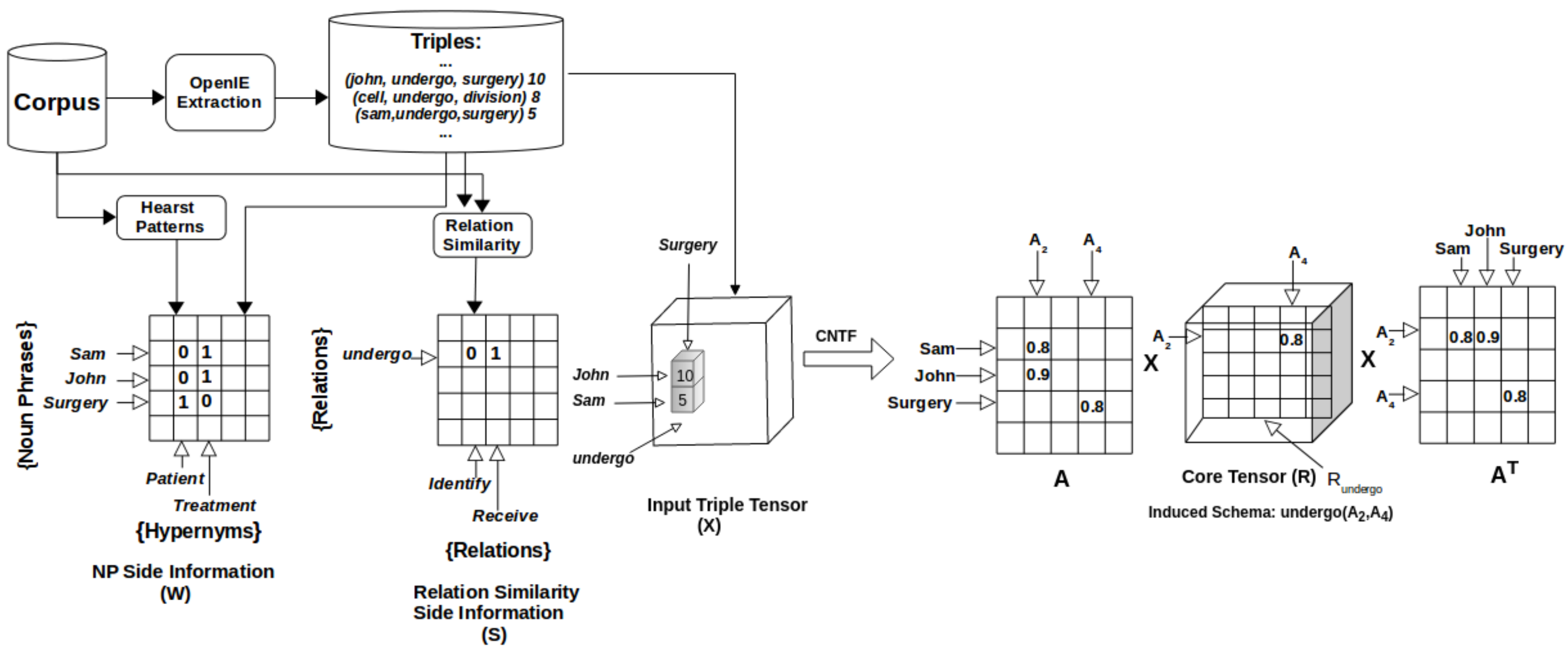


Figure : Plate Diagram of KB-LDA (figure taken from the original paper).

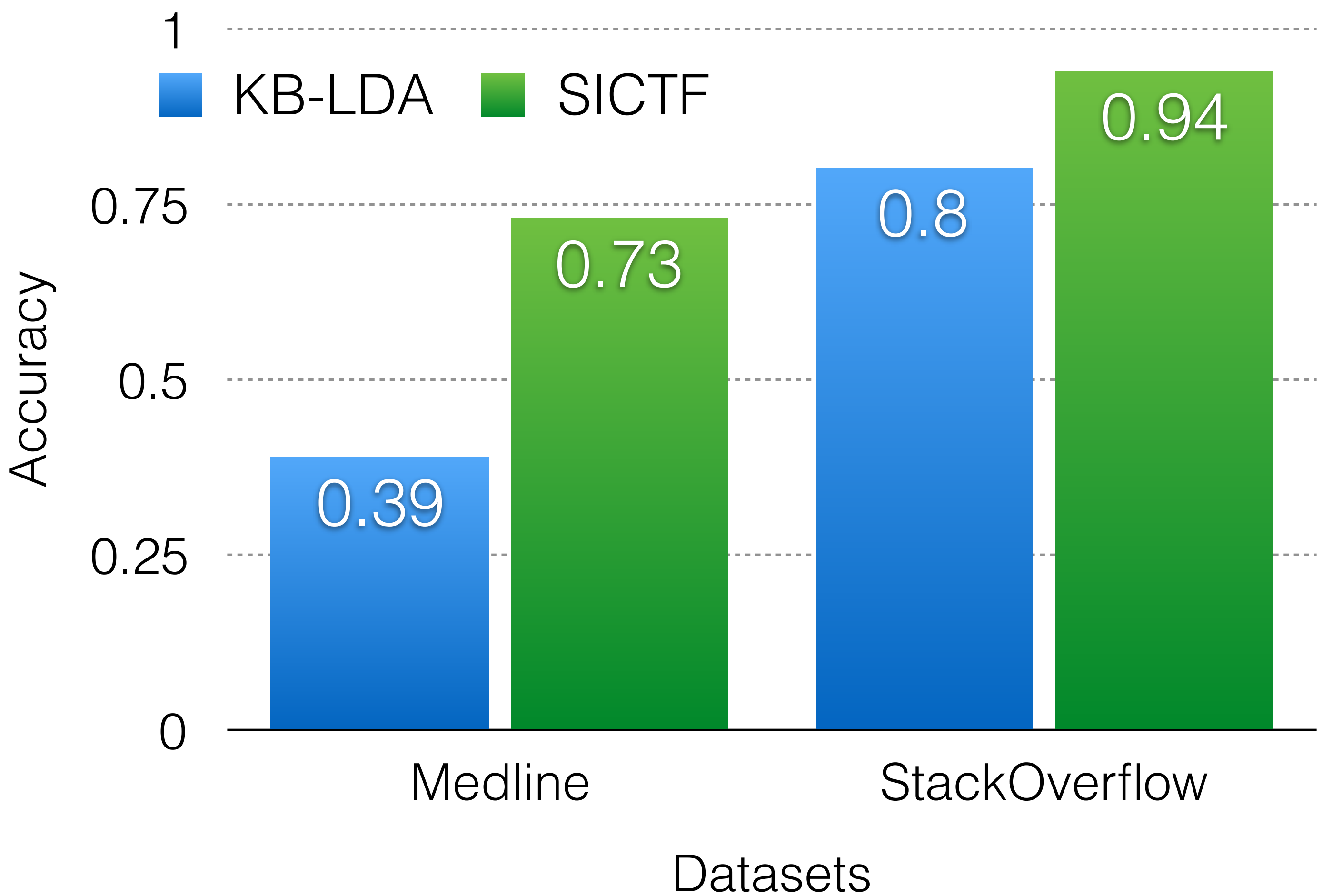
SICTF [Nimishakavi et al., 2016]

- Schema induction using coupled tensor-matrix factorization
- Inputs: SVO triples tensor, NP x Category side info matrix, relation similarity side info matrix

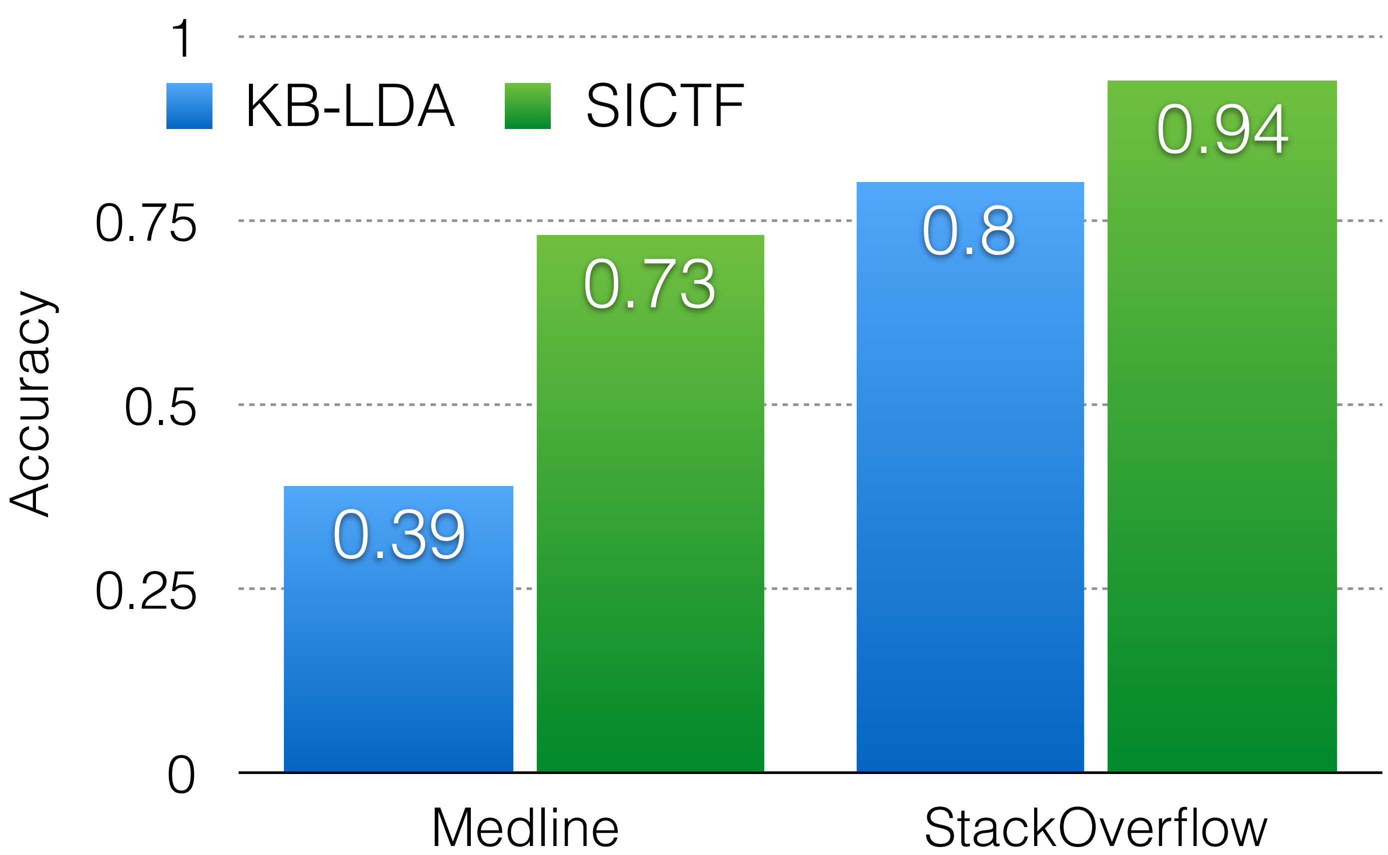


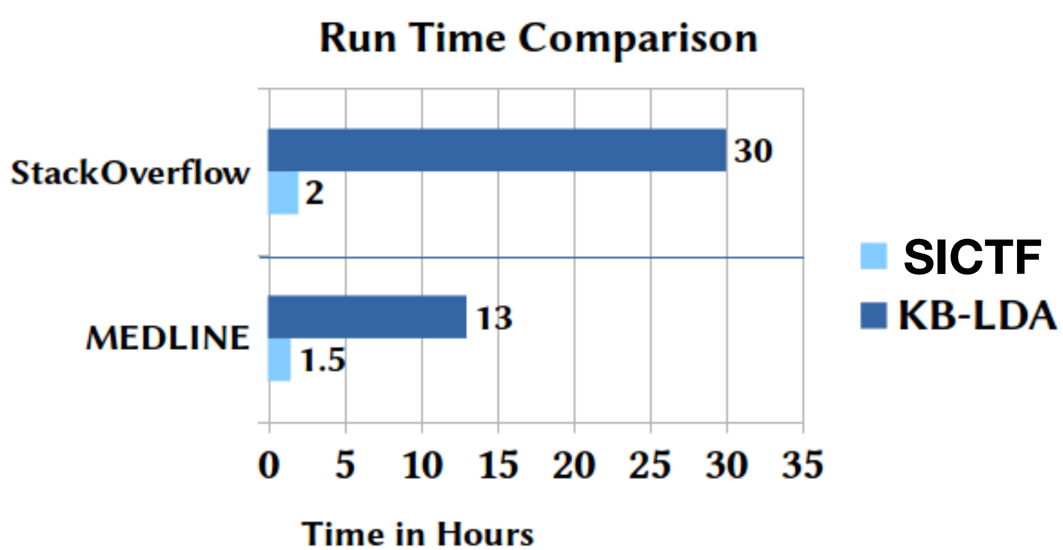
Binary Schema Induction Results

Binary Schema Induction Results

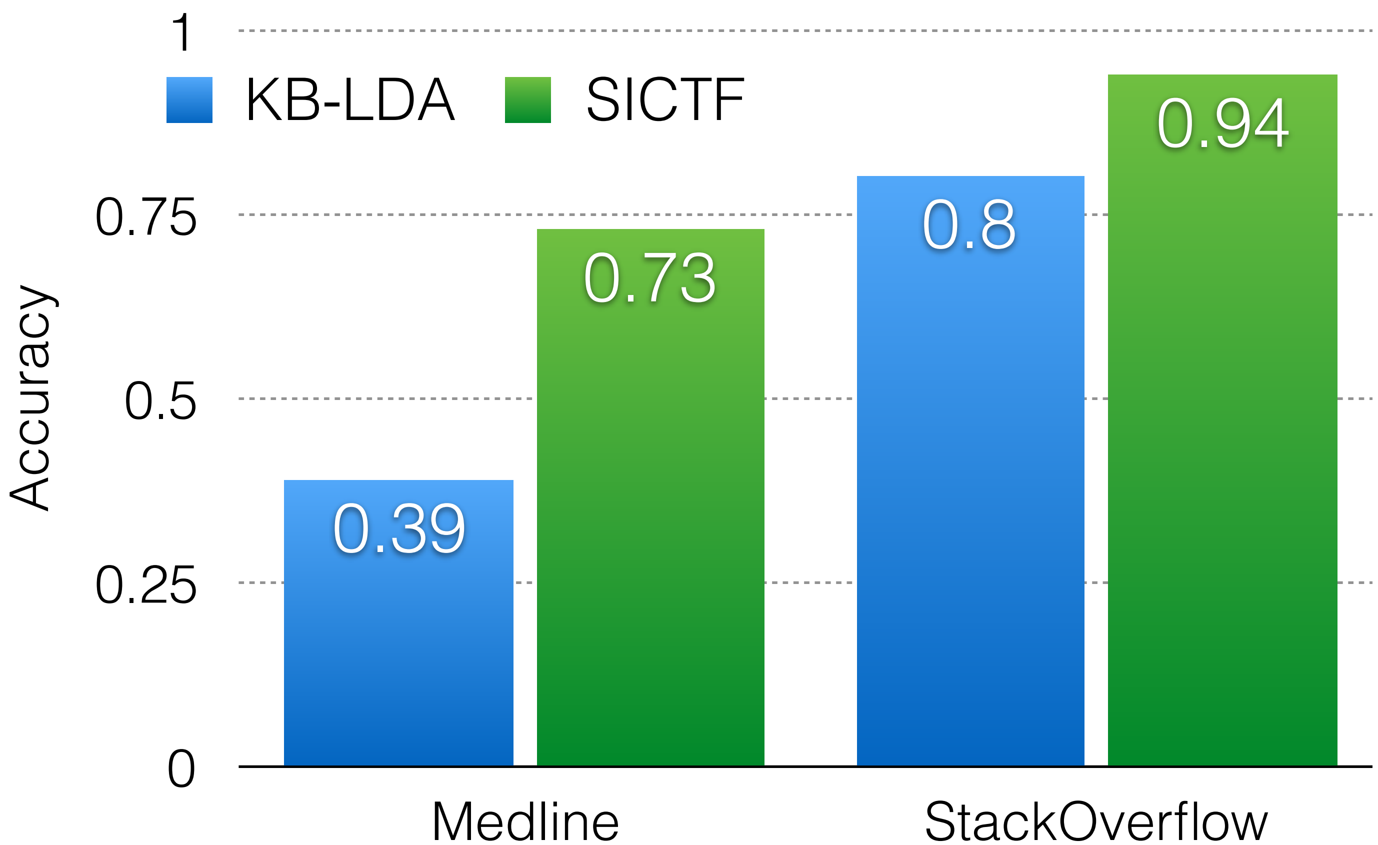


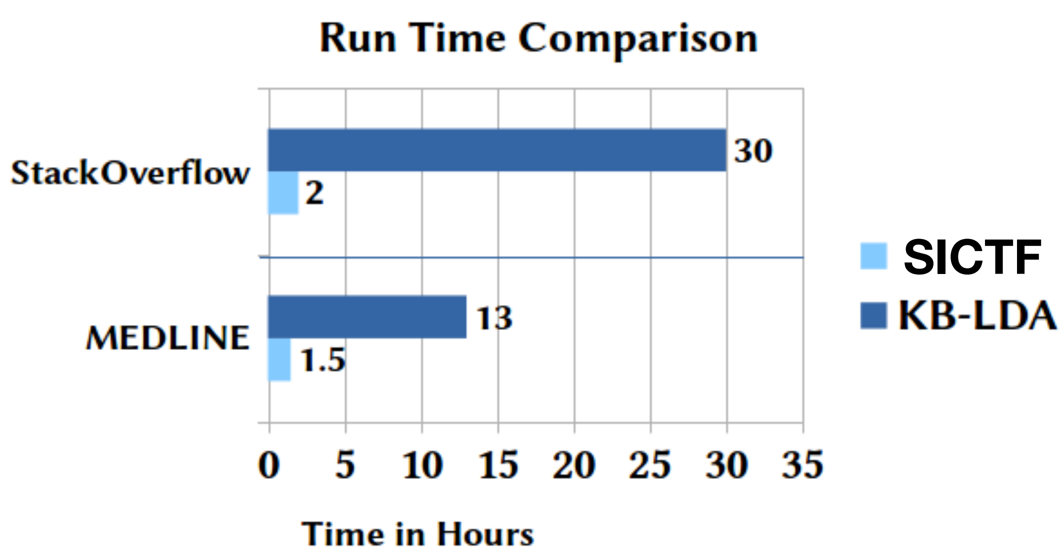
Binary Schema Induction Results





Binary Schema Induction Results

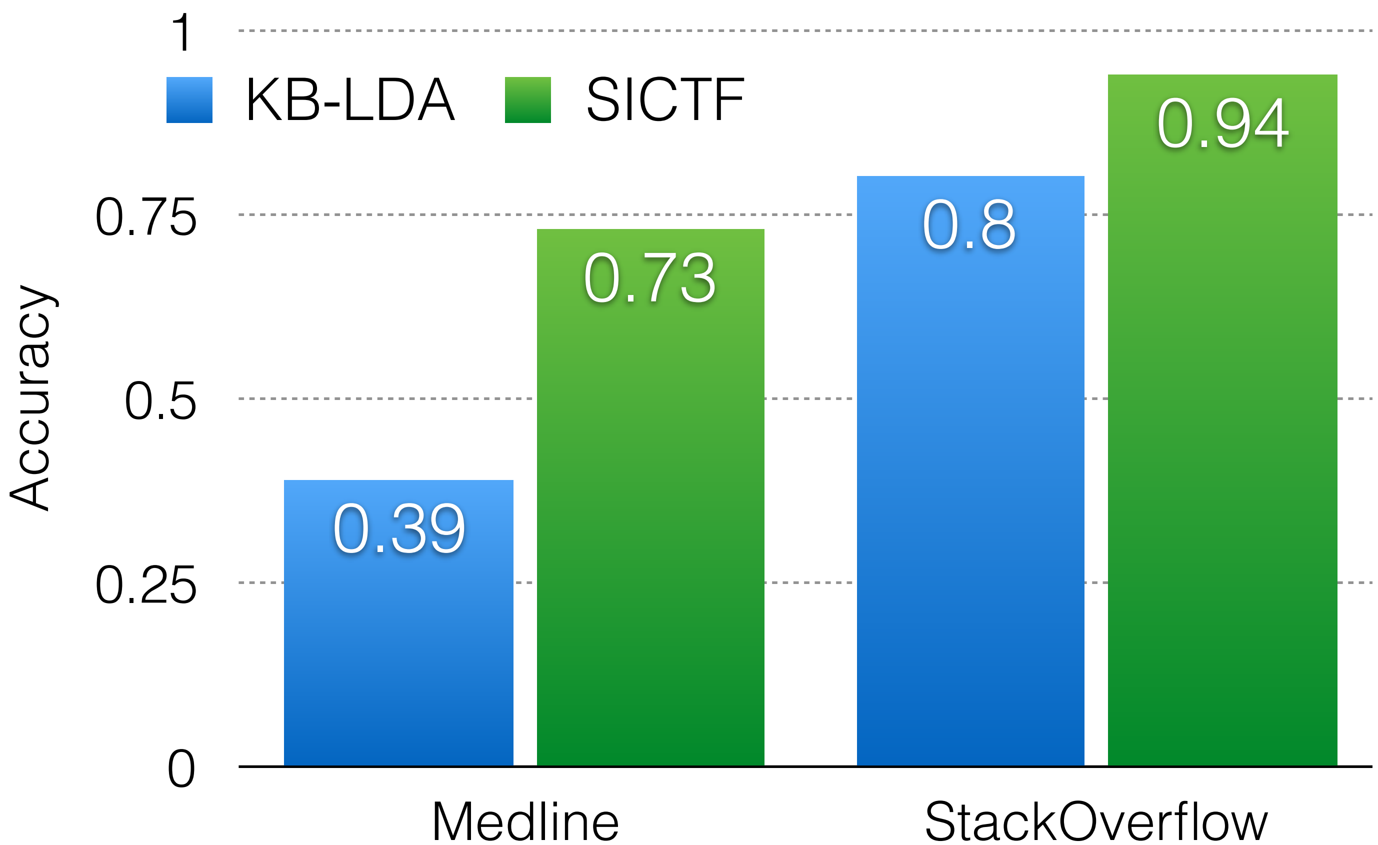


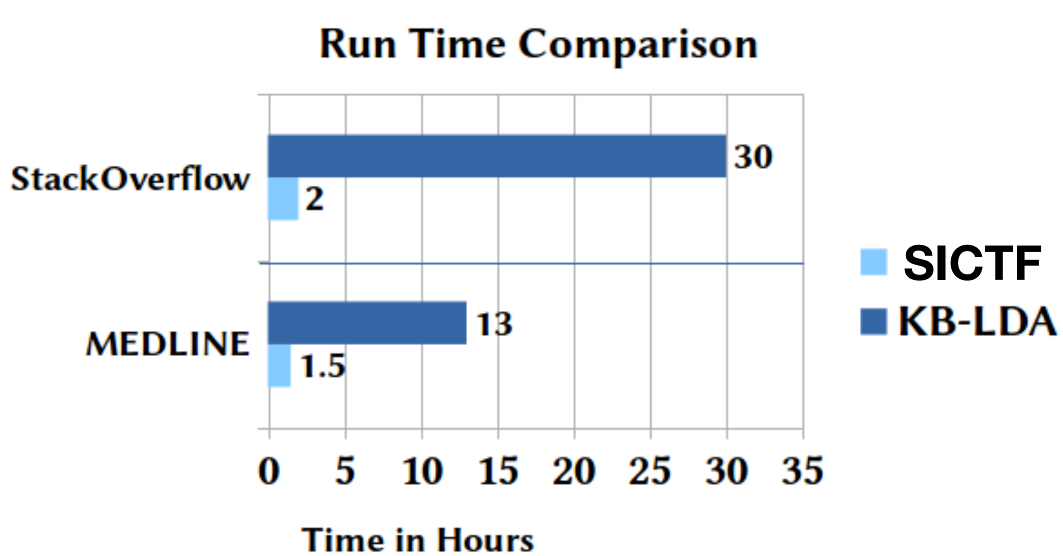


Relation Schema	Top 3 NPs in Induced Categories which were presented to annotators	Annotator Judgment
	StackOveflow	
clicks(A_0, A_1)	A_0: *users, client, person* A_1: *link, image, item*	valid
refreshes(A_{19}, A_{13})	A_{19}: *browser, window, tab* A_{13}: *page, activity, app*	valid
can_parse(A_{41}, A_{17})	A_{41}: *access, permission, ability* A_{17}: *image file, header file, zip file*	invalid
	MEDLINE	
receive(A_1, A_{18})	A_1: *patient, NUM patients, one patient* A_{18}: *flecainide, aerosolized pentamidine, prophylaxis*	valid
undergo(A_1, A_3)	A_1: *patient, NUM patients, one patient* A_3: *surgery, abdominal surgery, open heart surgery*	valid
fail_to(A_{32}, A_{36})	A_{32}: *chest pain, bacteriologic failure, unresectable disease* A_{36}: *nodular disease, valvular disease, Crohn disease*	invalid

SICTF induced schemas

Binary Schema Induction Results





Relation Schema	Top 3 NPs in Induced Categories which were presented to annotators	Annotator Judgment
	StackOveflow	
clicks(A_0, A_1)	A_0: *users, client, person* A_1: *link, image, item*	valid
refreshes(A_{19}, A_{13})	A_{19}: *browser, window, tab* A_{13}: *page, activity, app*	valid
can_parse(A_{41}, A_{17})	A_{41}: *access, permission, ability* A_{17}: *image file, header file, zip file*	invalid
	MEDLINE	
receive(A_1, A_{18})	A_1: *patient, NUM patients, one patient* A_{18}: *flecainide, aerosolized pentamidine, prophylaxis*	valid
undergo(A_1, A_3)	A_1: *patient, NUM patients, one patient* A_3: *surgery, abdominal surgery, open heart surgery*	valid
fail_to(A_{32}, A_{36})	A_{32}: *chest pain, bacteriologic failure, unresectable disease* A_{36}: *nodular disease, valvular disease, Crohn disease*	invalid

SICTF induced schemas

SICTF Code: https://github.com/malllabiisc/sictf

TFBA [Nimishakavi et al., 2018]

- Induces **higher-order** relation schemas (beyond binary)
- Factorizes higher-oder tensor by backing-off into multiple lower-order tensors, factorizes everything jointly

TFBA [Nimishakavi et al., 2018]

- Induces **higher-order** relation schemas (beyond binary)
- Factorizes higher-oder tensor by backing-off into multiple lower-order tensors, factorizes everything jointly

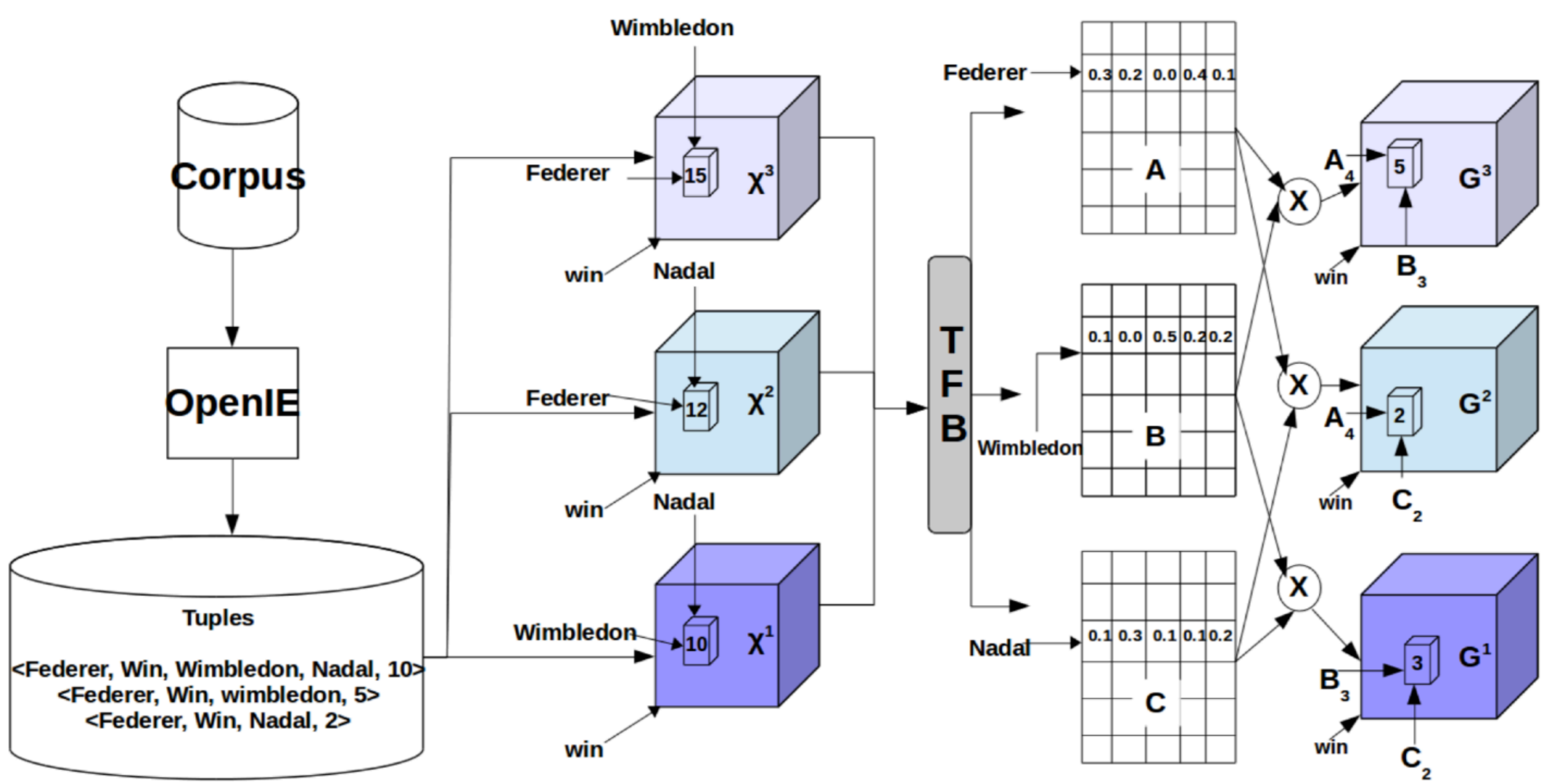
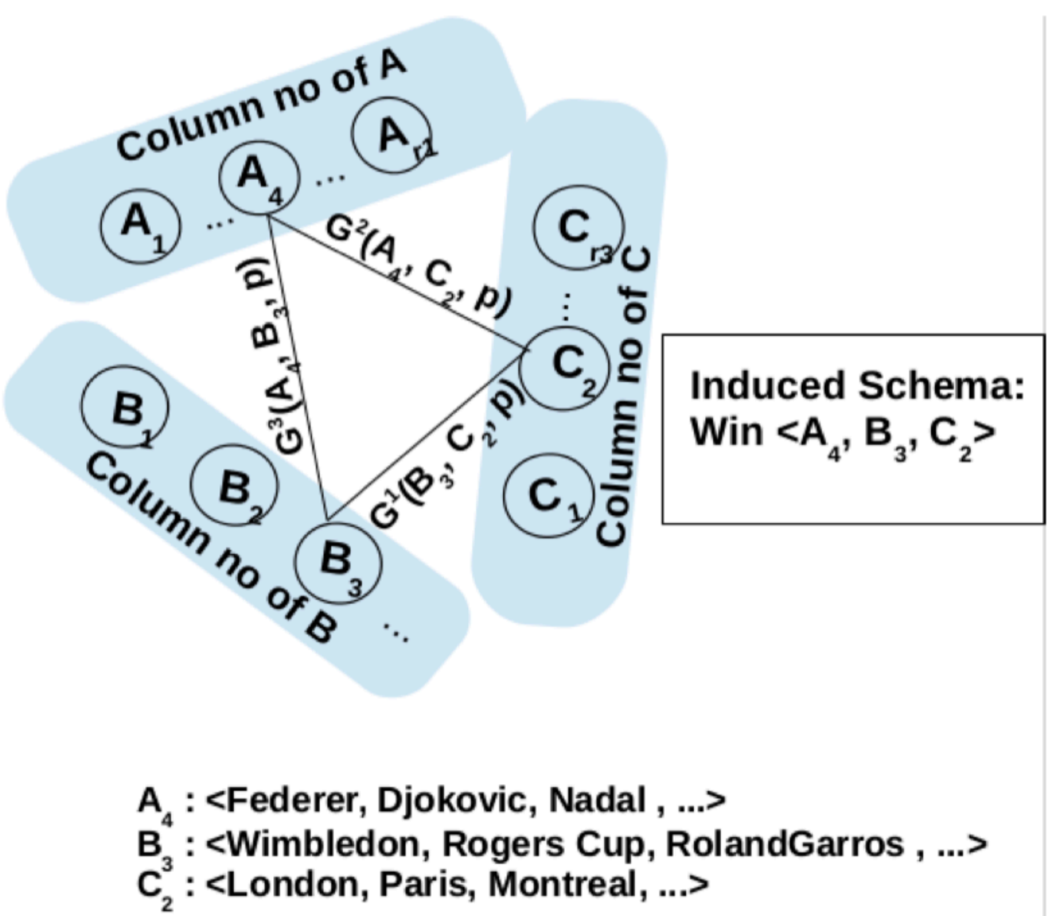


Figure : Tensor Factorization with Back-off

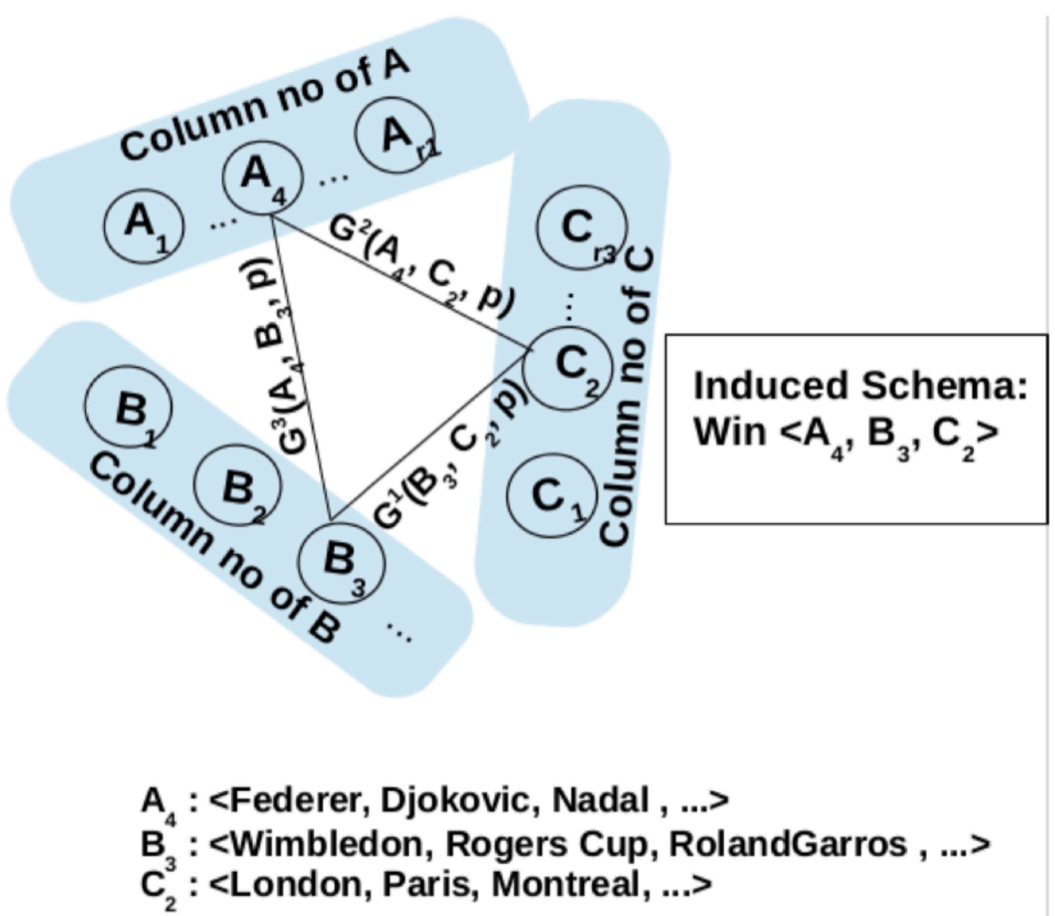
TFBA (contd.)

- TFBA constructs higher-order schemas by solving a constrained-clique mining



TFBA (contd.)

- TFBA constructs higher-order schemas by solving a constrained-clique mining

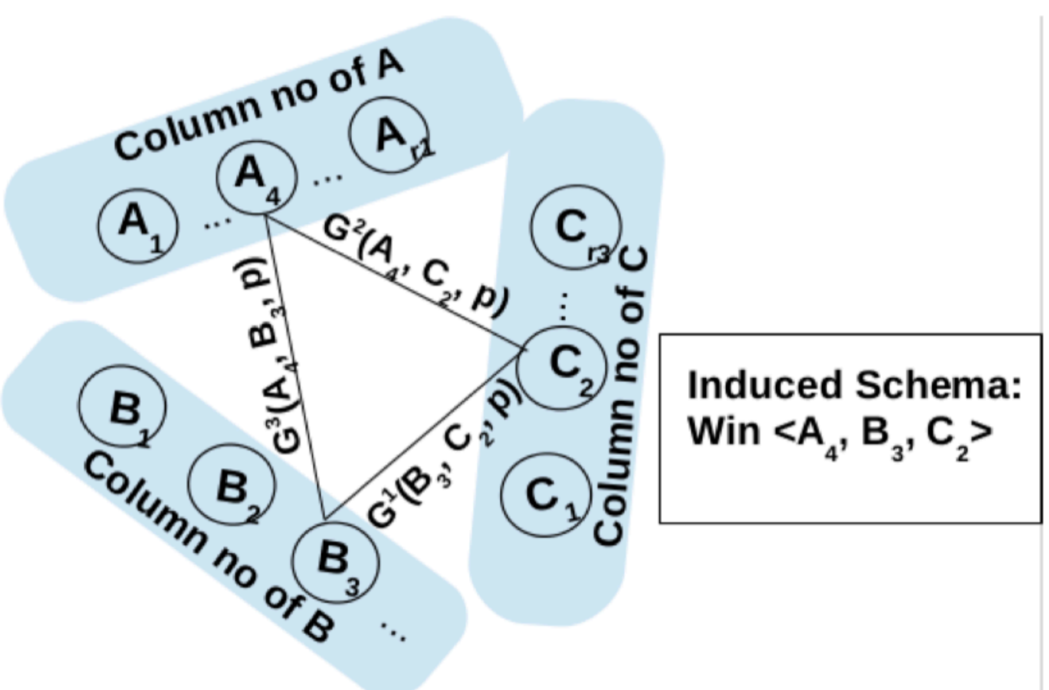


Relation Schema	NPs from the induced categories
	Shootings
leave(A_6, B_0, C_7)	A_6: *shooting, shooting incident, double shooting* B_0: *one person, two people, three people* C_7: *dead, injured, on edge*
identify(A_1, B_1, C_5, C_6)	A_1: *police, officers, huntsville police* B_1: *man, victims, four victims* C_5: *sunday, shooting staurday, wednesday afternoon* C_6: *apartment, bedroom, building in the neighborhood*
say(A_1, B_1, C_5)	A_1: *police, officers, huntsville police* B_1: *man, victims, four victims* C_5: *sunday, shooting staurday, wednesday afternoon*
	NYT sports
spend(A_0, B_{16}, C_3)	A_0: *yankees, mets, jets* B_{16}: *\$ <num> million, \$ <num>, \$ <num> billion* C_3: *<num>, year, last season*
win(A_2, B_{10}, C_3)	A_2: *red sox, team, yankees* B_{10}: *world series, title, world cup* C_3: *<num>, year, last season*
get(A_4, B_4, C_1)	A_4: *umpire, mike cameron, andre agassi* B_4: *ball, lives, grounder* C_1: *back, forward, <num>-yard line*

TFBA induced schemas

TFBA (contd.)

- TFBA constructs higher-order schemas by solving a constrained-clique mining



A_4 : <Federer, Djokovic, Nadal , ...>
B_3 : <Wimbledon, Rogers Cup, RolandGarros , ...>
C_2 : <London, Paris, Montreal, ...>

Relation Schema	NPs from the induced categories
	Shootings
leave(A_6, B_0, C_7)	A_6: *shooting, shooting incident, double shooting* B_0: *one person, two people, three people* C_7: *dead, injured, on edge*
identify(A_1, B_1, C_5, C_6)	A_1: *police, officers, huntsville police* B_1: *man, victims, four victims* C_5: *sunday, shooting staurday, wednesday afternoon* C_6: *apartment, bedroom, building in the neighborhood*
say(A_1, B_1, C_5)	A_1: *police, officers, huntsville police* B_1: *man, victims, four victims* C_5: *sunday, shooting staurday, wednesday afternoon*
	NYT sports
spend(A_0, B_{16}, C_3)	A_0: *yankees, mets, jets* B_{16}: *\$ <num> million, \$ <num>, \$ <num> billion* C_3: *<num>, year, last season*
win(A_2, B_{10}, C_3)	A_2: *red sox, team, yankees* B_{10}: *world series, title, world cup* C_3: *<num>, year, last season*
get(A_4, B_4, C_1)	A_4: *umpire, mike cameron, andre agassi* B_4: *ball, lives, grounder* C_1: *back, forward, <num>-yard line*

TFBA induced schemas

TFBA Code: https://github.com/madhavcsa/TFBA

Outline

13:00-13:15 Overview and motivation

13:15-13:45 Case study: NELL

13:45-14:00 Bootstrapped Entity Extraction

14:00-15:00 Open Relation Extraction & Canonicalization

15:00-15:30 **Coffee Break**

15:30-16:15 Distantly-supervised Relation Extraction

16:15-16:45 Knowledge Graph Embeddings

16:45-17:00 Conclusion & QA