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)

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: 31 years

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: 31 years

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
- Amazon Product 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
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:
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
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 Today


Recently-Learned Facts

<table>
<thead>
<tr>
<th>instance</th>
<th>iteration</th>
<th>date learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>mark_bellhorn is a Mexican person</td>
<td>763</td>
<td>27-aug-2013</td>
</tr>
<tr>
<td>methenamine_mandelate_tablet is a drug</td>
<td>763</td>
<td>27-aug-2013</td>
</tr>
<tr>
<td>pete_zimmer is a person</td>
<td>763</td>
<td>27-aug-2013</td>
</tr>
<tr>
<td>sandhills_clubtail is a vertebrate</td>
<td>764</td>
<td>31-aug-2013</td>
</tr>
<tr>
<td>jeffrey_carlson is a chef</td>
<td>763</td>
<td>27-aug-2013</td>
</tr>
<tr>
<td>sutton is a park in the city london</td>
<td>767</td>
<td>06-sep-2013</td>
</tr>
<tr>
<td>pushkin was born in moscow</td>
<td>767</td>
<td>06-sep-2013</td>
</tr>
<tr>
<td>honda is a company that produces accord</td>
<td>766</td>
<td>04-sep-2013</td>
</tr>
<tr>
<td>spurs is a sports team that plays against magic</td>
<td>763</td>
<td>27-aug-2013</td>
</tr>
<tr>
<td>baseball is a sport played in the venue ballpark_in_ Arlington</td>
<td>766</td>
<td>04-sep-2013</td>
</tr>
</tbody>
</table>
Other Related Efforts

Cycorp
Home of smarter solutions

Freebase
High Supervision

yago
select knowledge

NELL
Low Supervision

Google
“KNOWLEDGE VAULTS”
Never-Ending Learning


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 can 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 120m diverse, confidence-weighted beliefs (e.g., servedWith(tea,biscuits)), while learning thousands of interrelated functions that continually improve its reading comprehension 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://rws.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 such as 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 → Y (e.g., the spam filter) given a training set of input/output pairs {(x_i, y_i)} of that function. Other machine learning paradigms exist as well (e.g., unsupervised clustering, topic modeling, reinforcement learning) but these paradigms also typically require 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.2

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.
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 in tens of millions, right plot vertical axis is in millions.
NELL's Accuracy over Time

Precision of top 10 predictions

- Y-axis: precision
- X-axis: iterations
- Line graph showing precision over iterations
How does NELL work?
Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier
Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

mayor of arg1
live in arg1
Learn which noun phrases are cities:

Paris  San Francisco
Pittsburgh Berlin
Seattle denial
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

arg1 is home of traits such as arg1
Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

San Francisco
Berlin
denial

anxiety
selfishness
London

mayor of arg1
live in arg1

arg1 is home of traits such as arg1
Semi-Supervised Bootstrap Learning

Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

San Francisco
Berlin
denial

anxiety
selfishness
London

mayor of arg1
live in arg1

arg1 is home of traits such as arg1

it's underconstrained!!
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: \[ \sum_{<np,person> \in \text{labeled data}} |f_1(np) - person| \]

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

NP: NP context distribution

morphology

\[ \text{__ is a friend} \]
\[ \text{rang the __} \]
\[ \text{ends with ‘...ski’?} \]
\[ \text{__ walked in} \]
\[ \text{contains “univ.”?} \]
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

person

f₁(NP)
f₂(NP)
f₃(NP)

NP morphology

NP HTML contexts

— is a friend
rang the __
capitalized?
ends with ‘…ski’?
contains “univ.”?

www.celebrities.com:

<li> __ </li>

— walked in
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]
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) \implies \text{athlete}(NP1), \text{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_social_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_legends_as_arg1
  arg2_legends_such_as_arg1  arg2_players_is_arg1  arg2_pro_arg1
  arg2_player_was_arg1  arg2_great_arg1  arg2_hero_arg1
  arg1_was_born_to_play_arg2  arg2_star_arg1  arg2_hall_of_famer_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_megastar_arg1
arg2_player_named_arg1  arg2_prodigy_arg1
arg1_is_the_tiger_woods_of_arg2  arg2_greats_like_arg1
arg1_plays_arg2  arg2_player_is_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_greats_such_as_arg1  arg2_professionals_such_as_arg1
arg2_hit_by_arg1  arg2_greats_arg1
arg2_pros_like_arg1  arg1_retires_from_arg2
arg2_lesson_from_arg1  arg2_architects_robert_trent_jones_and_arg1
arg2_sensation_arg1  arg2_pros_arg1
arg2_hall_of_famer_arg1  arg2_superstar_arg1
arg2_legends_such_as_arg1  arg2_players_is_arg1
arg2_player_was_arg1  arg2_god_arg1
arg1_was_born_to_play_arg2  arg2_star_arg1
arg2_players_are_arg1  arg1_retired_from_professional_arg2
arg2_legends_as_arg1  arg2_autographed_by_arg1

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Feature</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>mountain</td>
<td>LAST=peak</td>
<td>1.791</td>
</tr>
<tr>
<td>mountain</td>
<td>LAST=mountain</td>
<td>1.093</td>
</tr>
<tr>
<td>mountain</td>
<td>FIRST=mountain</td>
<td>-0.875</td>
</tr>
<tr>
<td>musicArtist</td>
<td>LAST=band</td>
<td>1.853</td>
</tr>
<tr>
<td>musicArtist</td>
<td>POS=DT_NNS</td>
<td>1.412</td>
</tr>
<tr>
<td>musicArtist</td>
<td>POS=DT_JJ_NN</td>
<td>-0.807</td>
</tr>
<tr>
<td>newspaper</td>
<td>LAST=sun</td>
<td>1.330</td>
</tr>
<tr>
<td>newspaper</td>
<td>LAST=university</td>
<td>-0.318</td>
</tr>
<tr>
<td>newspaper</td>
<td>POS=NN_NNS</td>
<td>-0.798</td>
</tr>
<tr>
<td>university</td>
<td>LAST=college</td>
<td>2.076</td>
</tr>
<tr>
<td>university</td>
<td>PREFIX=uc</td>
<td>1.999</td>
</tr>
<tr>
<td>university</td>
<td>LAST=state</td>
<td>1.992</td>
</tr>
<tr>
<td>university</td>
<td>LAST=university</td>
<td>1.745</td>
</tr>
<tr>
<td>university</td>
<td>FIRST=college</td>
<td>-1.381</td>
</tr>
<tr>
<td>visualArtMovement</td>
<td>SUFFIX=ism</td>
<td>1.282</td>
</tr>
<tr>
<td>visualArtMovement</td>
<td>PREFIX=journ</td>
<td>-0.234</td>
</tr>
<tr>
<td>visualArtMovement</td>
<td>PREFIX=budd</td>
<td>-0.253</td>
</tr>
</tbody>
</table>
NELL: Learned reading strategies

Plays_Sport(arg1, arg2):
- arg1_was_playing_arg2
- arg2_megastar
- arg2_icon
- arg2_great
- arg2_player_named_arg1
- arg2_prodigy
- arg1_is_the_tiger_woods_of_arg2
- arg2_career_of_arg1
- arg2_legends
- arg1_announced_his_retirement_from_arg2
- arg2_operations_chief
- 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_greats_such_as_arg1
- arg2_professionals_such_as_arg1
- arg2_hit_by_arg1
- arg2_greats
- arg2_icon
- arg2_stars_like_arg1
- arg2_pros_like_arg1
- arg1_retires_from_arg2
- arg2_phenom
- arg2_lesson_from_arg1
- arg2_architect
- arg2_sensation_arg1
- arg2_pros_arg1
- arg2_stars_venus_and_arg1
- arg2_hall_of_famer_arg1
- arg2_superstar
- arg2_legends_such_as_arg1
- arg2_player_was_arg1
- arg2_god_arg1
- arg1_was_born_to_play_arg2
- arg2_star
- arg2_hero
- arg2_players_are_arg1
- arg1_retired_from_professional_arg2
- arg2_legends_as_arg1
- arg2_autographed_by_arg1
- arg2_champion

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

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Web URL</th>
<th>Extraction Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>academicField</td>
<td><a href="http://scholendow.ais.msu.edu/student/ScholSearch.Asp">http://scholendow.ais.msu.edu/student/ScholSearch.Asp</a></td>
<td> [X] -</td>
</tr>
</tbody>
</table>
| bird         | http://www.michaelforsberg.com/stock.html      | &lt;option&gt;[X]&lt;/option&gt;
NELL Architecture

Knowledge Base (latent variables)
- Beliefs
- Candidate Beliefs

Evidence Integrator

Text Context patterns (CPL)
- Actively search for web text (OpenEval)

Orthographic classifier (CML)
- Infer new beliefs from old (PRA)

URL specific HTML patterns (SEAL)
- Image classifier (NEIL)

Human advice
- Ontology extender (OntExt)
If coupled learning is the key, how can we get new coupling constraints?
Learned Probabilistic Horn Clause Rules

0.93 \( \text{playsSport}(x,y) \leftarrow \text{playsForTeam}(x,z), \text{teamPlaysSport}(z,y) \)
Learned Probabilistic Horn Clause Rules

0.93 \( \text{playsSport}(?x,?y) \leftarrow \text{playsForTeam}(?x,?z), \text{teamPlaysSport}(?z,?y) \)
Inference by KB Random Walks

KB:

Random walk path type: $x \xrightarrow{competes\ with} ? \xrightarrow{economic\ sector} y$

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 $i$
<table>
<thead>
<tr>
<th>Feature = Typed Path</th>
<th>Feature Value</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityInState, CityInstate^−1, CityLocatedInCountry</td>
<td>0.8</td>
<td>0.32</td>
</tr>
</tbody>
</table>
CityLocatedInCountry(Pittsburgh) = ?

Feature = Typed Path
CityInState, CityInstate^{-1}, CityLocatedInCountry

Feature Value
0.8

Logistic Regression Weight
0.32

Pittsburgh
Pennsylvania

CityInState
CityLocatedInCountry(Pittsburgh) = ?

[[Lao et al, EMNLP 2011]]

**Feature = Typed Path**

CityInState, CityInState⁻¹, CityLocatedInCountry

**Feature Value**

<table>
<thead>
<tr>
<th>Feature Value</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.32</td>
</tr>
</tbody>
</table>

---

**Logistic Regression Weight**

---
CityLocatedInCountry(Pittsburgh) = ?

[Feature = Typed Path]
CityInState, CityInState^{-1}, CityLocatedInCountry

[Feature Value]
0.8

[Logistic Regression Weight]
0.32

[Lao et al, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

Pr(U.S. | Pittsburgh, TypedPath)

Feature = Typed Path
CityInState, CityInstate$^{-1}$, CityLocatedInCountry

Feature Value
0.8

Logistic Regression
Weight
0.32

[Lao et al, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

Feature = Typed Path
CityInState, CityInState⁻¹, CityLocatedInCountry
AtLocation⁻¹, AtLocation, CityLocatedInCountry

Feature Value
CityLocatedInCountry(Pittsburgh) = 0.8

Logistic Regression Weight
0.32
0.20

[Laò et al, EMNLP 2011]
Feature = Typed Path
CityInState, CityInState⁻¹, CityLocatedInCountry
AtLocation⁻¹, AtLocation, CityLocatedInCountry

Feature Value
CityLocatedInCountry(Pittsburgh) = ?

Logistic Regression
Weight
0.8 0.32
0.20

[Lao et al, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

- Feature = Typed Path
  - CityInState, CityInState⁻¹, CityLocatedInCountry
  - AtLocation⁻¹, AtLocation, CityLocatedInCountry

Feature Value
- CityLocatedInCountry: 0.8, Weight: 0.32
- AtLocation: 0.20

[Laos et al, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

Feature = Typed Path
CityInState, CityInState⁻¹, CityLocatedInCountry
AtLocation⁻¹, AtLocation, CityLocatedInCountry

Feature Value
0.8
0.6

Weight
0.32
0.20

[Logistic Regression
Lao et al, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = U.S.  p=0.58

Feature = Typed Path
CityInState, CityInState^{-1}, CityLocatedInCountry
AtLocation^{-1}, AtLocation, CityLocatedInCountry
...

Feature Value
0.8  0.32
0.6  0.20
...
...

Logistic Regression Weight
Random walk inference: learned path types

CityLocatedInCountry(\textit{city}, \textit{country}): 

8.04 \textit{cityliesonriver}, \textit{cityliesonriver}^{-1}, \textit{citylocatedincountry} 
5.42 \textit{hasofficeincity}^{-1}, \textit{hasofficeincity}, \textit{citylocatedincountry} 
4.98 \textit{cityalsoknownas}, \textit{cityalsoknownas}, \textit{citylocatedincountry} 
2.85 \textit{citycapitalofcountry}, \textit{citylocatedincountry}^{-1}, \textit{citylocatedincountry} 
2.29 \textit{agentactsinlocation}^{-1}, \textit{agentactsinlocation}, \textit{citylocatedincountry} 
1.22 \textit{statehascapital}^{-1}, \textit{statelocatedincountry} 
0.66 \textit{citycapitalofcountry} 

7 of the 2985 learned paths for \textit{CityLocatedInCountry}
Key Idea 3: Automatically extend ontology
<table>
<thead>
<tr>
<th>Category Pair</th>
<th>Frequent Instance Pairs</th>
<th>Text Contexts</th>
<th>Suggested Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MusicInstrument</td>
<td>sitar, George Harrison</td>
<td>ARG1 master ARG2</td>
<td>Master</td>
</tr>
<tr>
<td>Musician</td>
<td>tenor sax, Stan Getz</td>
<td>ARG1 virtuoso ARG2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>trombone, Tommy Dorsey</td>
<td>ARG1 legend ARG2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>vibes, Lionel Hampton</td>
<td>ARG2 plays ARG1</td>
<td></td>
</tr>
<tr>
<td>Disease</td>
<td>pinched nerve, herniated disk</td>
<td>ARG1 is due to ARG2</td>
<td>IsDueTo</td>
</tr>
<tr>
<td></td>
<td>tennis elbow, tendonitis</td>
<td>ARG1 is caused by ARG2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>blepharospasm, dystonia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CellType</td>
<td>epithelial cells, surfactant</td>
<td>ARG1 that release ARG2</td>
<td>ThatRelease</td>
</tr>
<tr>
<td>Chemical</td>
<td>neurons, serotonin</td>
<td>ARG2 releasing ARG1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mast cells, histomine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mammals</td>
<td>koala bears, eucalyptus</td>
<td>ARG1 eat ARG2</td>
<td>Eat</td>
</tr>
<tr>
<td>Plant</td>
<td>sheep, grasses</td>
<td>ARG2 eating ARG1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>goats, saplings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>River</td>
<td>Seine, Paris</td>
<td>ARG1 in heart of ARG2</td>
<td>InHeartOf</td>
</tr>
<tr>
<td>City</td>
<td>Nile, Cairo</td>
<td>ARG1 which flows through ARG2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tiber river, Rome</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Mohamed et al. EMNLP 2011]
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, …
Seed Unlabeled Data

Extract Context

Induce Patterns

Patterns as Extractor

Extended List

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
...

Context Pattern Induction
**Extractions using Context Patterns**

### Induced Patterns (containing sequence "watch")

<table>
<thead>
<tr>
<th>Pattern 1</th>
<th>Pattern 2</th>
<th>Pattern 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>gold -&lt;ENT&gt;- watch</td>
<td>Richemont, -&lt;ENT&gt;- watches</td>
<td>Rolex watches are sold through official -&lt;ENT&gt;- and bought -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>diamond -&lt;ENT&gt;- watch</td>
<td>bought -&lt;ENT&gt;- watches</td>
<td>bought a -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>fake -&lt;ENT&gt;- watches</td>
<td>fake -&lt;ENT&gt;- watch</td>
<td>watchmaker -&lt;ENT&gt;- SA</td>
</tr>
<tr>
<td>bought -&lt;ENT&gt;- watch</td>
<td>diamond -&lt;ENT&gt;- watches</td>
<td>Ulysse -&lt;ENT&gt;- watches</td>
</tr>
<tr>
<td>encrusted -&lt;ENT&gt;- watch</td>
<td>stole -&lt;ENT&gt;- watches</td>
<td>Rolex watches and -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>stole -&lt;ENT&gt;- watch</td>
<td>buy a -&lt;ENT&gt;- watch</td>
<td>Rolex, -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>Richemont AG, -&lt;ENT&gt;- watches</td>
<td>jewelry, including -&lt;ENT&gt;- watch</td>
<td>Rolex and -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>Rolex and -&lt;ENT&gt;- watches</td>
<td>watchmaker -&lt;ENT&gt;-</td>
<td>diamond - studded -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>buy -&lt;ENT&gt;- watches</td>
<td>jewelry, including -&lt;ENT&gt;- watches</td>
<td>diamond - encrusted -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>Cartier and -&lt;ENT&gt;- watches</td>
<td>stole a -&lt;ENT&gt;- watch</td>
<td>Cartier, and -&lt;ENT&gt;- watches</td>
</tr>
<tr>
<td>buy -&lt;ENT&gt;- watch</td>
<td>Rolex watches and -&lt;ENT&gt;-</td>
<td>buy a -&lt;ENT&gt;- watches</td>
</tr>
<tr>
<td>gold -&lt;ENT&gt;- watches</td>
<td>watchmaker -&lt;ENT&gt;- Group</td>
<td>bought a -&lt;ENT&gt;- watches</td>
</tr>
</tbody>
</table>

### Extracted Lists Improve NER Taggers

<table>
<thead>
<tr>
<th>Training Data (Tokens)</th>
<th>Test-a No List</th>
<th>Test-a Seed List</th>
<th>Test-a Unsup. List</th>
</tr>
</thead>
<tbody>
<tr>
<td>9229</td>
<td>68.27</td>
<td>70.93</td>
<td><strong>72.26</strong></td>
</tr>
<tr>
<td>204657</td>
<td>89.52</td>
<td>84.30</td>
<td><strong>90.48</strong></td>
</tr>
</tbody>
</table>
Extractions using Context Patterns

Induced Patterns (containing sequence "watch")

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Example</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>gold -&lt;ENT&gt;- watch</td>
<td>Richemont, -&lt;ENT&gt;- watches</td>
<td>Rolex watches are sold through official -&lt;ENT&gt;- and</td>
</tr>
<tr>
<td>diamond -&lt;ENT&gt;- watch</td>
<td>bought -&lt;ENT&gt;- watches</td>
<td>bought a -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>fake -&lt;ENT&gt;- watches</td>
<td>fake -&lt;ENT&gt;- watch</td>
<td>watchmaker -&lt;ENT&gt;- SA</td>
</tr>
<tr>
<td>bought -&lt;ENT&gt;- watch</td>
<td>diamond -&lt;ENT&gt;- watches</td>
<td>Ulysse -&lt;ENT&gt;- watches</td>
</tr>
<tr>
<td>encrusted -&lt;ENT&gt;- watch</td>
<td>stole -&lt;ENT&gt;- watches</td>
<td>Rolex watches and -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>stolen -&lt;ENT&gt;- watch</td>
<td>buy a -&lt;ENT&gt;- watch</td>
<td>Rolex -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>Richemont AG, -&lt;ENT&gt;- watches</td>
<td>jewelry, including -&lt;ENT&gt;- watch</td>
<td>Rolex and -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>Rolex and -&lt;ENT&gt;- watches</td>
<td>watchmaker -&lt;ENT&gt;-</td>
<td>diamond - studded -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>buy -&lt;ENT&gt;- watches</td>
<td>jewelry, including -&lt;ENT&gt;- watches</td>
<td>diamond - encrusted -&lt;ENT&gt;- watch</td>
</tr>
<tr>
<td>Cartier and -&lt;ENT&gt;- watches</td>
<td>stole a -&lt;ENT&gt;- watch</td>
<td>Cartier, and -&lt;ENT&gt;- watches</td>
</tr>
<tr>
<td>buy -&lt;ENT&gt;- watch</td>
<td>Rolex watches and -&lt;ENT&gt;-</td>
<td>buy a -&lt;ENT&gt;- watches</td>
</tr>
<tr>
<td>gold -&lt;ENT&gt;- watches</td>
<td>watchmaker -&lt;ENT&gt;- Group</td>
<td>bought a -&lt;ENT&gt;- watches</td>
</tr>
</tbody>
</table>

Entities Extracted by Above Patterns (ranked)

<table>
<thead>
<tr>
<th>Entities (most confident)</th>
<th>Fossil</th>
<th>Tag Heuer</th>
<th>Super Bowl</th>
<th>SPOT</th>
<th>Sekonda</th>
<th>Rolexes</th>
<th>Harry Winston</th>
<th>Hampton Spirit</th>
<th>Girard Perregaux</th>
<th>Frank Mueller</th>
<th>David Yurman</th>
<th>Chopard</th>
<th>Frank Mueller</th>
<th>NFL (least confident)</th>
</tr>
</thead>
</table>

Extracted Lists Improve NER Taggers

<table>
<thead>
<tr>
<th>Training Data (Tokens)</th>
<th>No List</th>
<th>Seed List</th>
<th>Unsup. List</th>
</tr>
</thead>
<tbody>
<tr>
<td>9229</td>
<td>68.27</td>
<td>70.93</td>
<td>72.26</td>
</tr>
<tr>
<td>204657</td>
<td>89.52</td>
<td>84.30</td>
<td>90.48</td>
</tr>
</tbody>
</table>
SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

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]

- **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

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

Figure 1. Flow chart of the SEAL system
SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

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

Figure 1. Flow chart of the SEAL system
SEAL: Set Expansion using the Web

[Wang and Cohen, ICDM 2007]

- 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]

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

- 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
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)
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

• 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)

<table>
<thead>
<tr>
<th>Source Type</th>
<th>Edge Relation</th>
<th>Target Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>seeds</td>
<td>find</td>
<td>document</td>
</tr>
<tr>
<td>document</td>
<td>derive</td>
<td>wrapper</td>
</tr>
<tr>
<td></td>
<td>find$^{-1}$</td>
<td>seeds</td>
</tr>
<tr>
<td>wrapper</td>
<td>extract</td>
<td>mention</td>
</tr>
<tr>
<td></td>
<td>derive$^{-1}$</td>
<td>document</td>
</tr>
<tr>
<td>mention</td>
<td>extract$^{-1}$</td>
<td>wrapper</td>
</tr>
</tbody>
</table>

Minkov et al. *Contextual Search and Name Disambiguation in Email using Graphs*. SIGIR 2006
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)
Top three mentions are the seeds:

<table>
<thead>
<tr>
<th>#</th>
<th>Entity</th>
<th>#</th>
<th>Entity</th>
<th>#</th>
<th>Entity</th>
<th>#</th>
<th>Entity</th>
<th>#</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>kdd</td>
<td>2</td>
<td>icml</td>
<td>3</td>
<td>icdm</td>
<td>4</td>
<td>ijcai</td>
<td>5</td>
<td>aai</td>
</tr>
<tr>
<td>2</td>
<td>andrew mccallum</td>
<td>2</td>
<td>amazing race</td>
<td>3</td>
<td>john lafferty</td>
<td>4</td>
<td>naftali tishby</td>
<td>5</td>
<td>fernando parreira</td>
</tr>
<tr>
<td>3</td>
<td>survivor</td>
<td>4</td>
<td>big brother</td>
<td>5</td>
<td>the apprentice</td>
<td>6</td>
<td>the bachelor</td>
<td>7</td>
<td>daphne koller</td>
</tr>
<tr>
<td>4</td>
<td>sam's club</td>
<td>5</td>
<td>sears</td>
<td>6</td>
<td>circuit city</td>
<td>7</td>
<td>best buy</td>
<td>8</td>
<td>joe millionaire</td>
</tr>
<tr>
<td>5</td>
<td>dhl</td>
<td>6</td>
<td>the mole</td>
<td>7</td>
<td>the swiss</td>
<td>8</td>
<td>average joe</td>
<td>9</td>
<td>office depot</td>
</tr>
<tr>
<td>6</td>
<td>sigir</td>
<td>7</td>
<td>thomas hofmann</td>
<td>8</td>
<td>the mole</td>
<td>9</td>
<td>sears</td>
<td>10</td>
<td>ace hardware</td>
</tr>
<tr>
<td>7</td>
<td>reality tv</td>
<td>8</td>
<td>sigir</td>
<td>9</td>
<td>the mole</td>
<td>10</td>
<td>kmart</td>
<td>11</td>
<td>thomas hofmann</td>
</tr>
<tr>
<td>8</td>
<td>dan roth</td>
<td>9</td>
<td>the mole</td>
<td>10</td>
<td>sigir</td>
<td>11</td>
<td>drugstore.com</td>
<td>12</td>
<td>thomas hofmann</td>
</tr>
<tr>
<td>9</td>
<td>ida</td>
<td>10</td>
<td>reality tv</td>
<td>11</td>
<td>the mole</td>
<td>12</td>
<td>sephora</td>
<td>13</td>
<td>the sports authority</td>
</tr>
<tr>
<td>10</td>
<td>william w. cohen</td>
<td>11</td>
<td>reality tv</td>
<td>12</td>
<td>the mole</td>
<td>13</td>
<td>the sports authority</td>
<td>14</td>
<td>the sports authority</td>
</tr>
<tr>
<td>11</td>
<td>uai</td>
<td>12</td>
<td>reality tv</td>
<td>13</td>
<td>the mole</td>
<td>14</td>
<td>the sports authority</td>
<td>15</td>
<td>the sports authority</td>
</tr>
<tr>
<td>12</td>
<td>ilp</td>
<td>13</td>
<td>the mole</td>
<td>14</td>
<td>the mole</td>
<td>15</td>
<td>the sports authority</td>
<td>16</td>
<td>the sports authority</td>
</tr>
<tr>
<td>13</td>
<td>stoc</td>
<td>14</td>
<td>the mole</td>
<td>15</td>
<td>the mole</td>
<td>16</td>
<td>the sports authority</td>
<td>17</td>
<td>the sports authority</td>
</tr>
<tr>
<td>14</td>
<td>yoram singer</td>
<td>15</td>
<td>the mole</td>
<td>16</td>
<td>the mole</td>
<td>17</td>
<td>the sports authority</td>
<td>18</td>
<td>the sports authority</td>
</tr>
<tr>
<td>15</td>
<td>www</td>
<td>16</td>
<td>the mole</td>
<td>17</td>
<td>the mole</td>
<td>18</td>
<td>the sports authority</td>
<td>19</td>
<td>the sports authority</td>
</tr>
<tr>
<td>16</td>
<td>michael i. jordan</td>
<td>17</td>
<td>the mole</td>
<td>18</td>
<td>the mole</td>
<td>19</td>
<td>the sports authority</td>
<td>20</td>
<td>the sports authority</td>
</tr>
<tr>
<td>17</td>
<td>icde</td>
<td>18</td>
<td>the mole</td>
<td>19</td>
<td>the mole</td>
<td>20</td>
<td>the sports authority</td>
<td>21</td>
<td>the sports authority</td>
</tr>
<tr>
<td>18</td>
<td>eugene charniak</td>
<td>19</td>
<td>the mole</td>
<td>20</td>
<td>the mole</td>
<td>21</td>
<td>the sports authority</td>
<td>22</td>
<td>the sports authority</td>
</tr>
<tr>
<td>19</td>
<td>amir globerson</td>
<td>20</td>
<td>the mole</td>
<td>21</td>
<td>the mole</td>
<td>22</td>
<td>the sports authority</td>
<td>23</td>
<td>the sports authority</td>
</tr>
<tr>
<td>20</td>
<td>yiming yang</td>
<td>21</td>
<td>the mole</td>
<td>22</td>
<td>the mole</td>
<td>23</td>
<td>the sports authority</td>
<td>24</td>
<td>the sports authority</td>
</tr>
<tr>
<td>21</td>
<td>temptation</td>
<td>22</td>
<td>the mole</td>
<td>23</td>
<td>the mole</td>
<td>24</td>
<td>the sports authority</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 ....

Extractions from HTML lists and tables

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

---

**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]

<table>
<thead>
<tr>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Dylan</td>
<td>Billy Joel</td>
</tr>
<tr>
<td>Johnny Cash</td>
<td>Johnny Cash</td>
</tr>
<tr>
<td>Billy Joel</td>
<td></td>
</tr>
</tbody>
</table>
Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]

<table>
<thead>
<tr>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Dylan</td>
<td>Billy Joel</td>
</tr>
<tr>
<td>Johnny Cash</td>
<td>Johnny Cash</td>
</tr>
<tr>
<td>Billy Joel</td>
<td></td>
</tr>
</tbody>
</table>
Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]

Set 1
Bob Dylan (0.95)
Johnny Cash
Billy Joel

Set 2
Billy Joel (0.72)
Johnny Cash

Extraction Confidence
Table Mining
Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]

Set 1
- Bob Dylan (0.95)
- Johnny Cash
- Billy Joel

Set 2
- Billy Joel (0.72)
- Johnny Cash

Extraction Confidence

Table Mining

Pattern
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]

<table>
<thead>
<tr>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Dylan</td>
<td>Billy Joel</td>
</tr>
<tr>
<td>Johnny Cash</td>
<td>Johnny Cash</td>
</tr>
<tr>
<td>Billy Joel</td>
<td></td>
</tr>
</tbody>
</table>

Extraction Confidence

Table Mining

Pattern

Bob Dylan
Johnny Cash
Billy Joel
Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]

<table>
<thead>
<tr>
<th>Set 1</th>
<th>Extraction Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Dylan</td>
<td>0.95</td>
</tr>
<tr>
<td>Johnny Cash</td>
<td>0.87</td>
</tr>
<tr>
<td>Billy Joel</td>
<td>0.82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billy Joel</td>
</tr>
<tr>
<td>Johnny Cash</td>
</tr>
</tbody>
</table>

Set 1

Set 2

Pattern

Table Mining

Extraction Confidence
Class-instance Acquisition using Graph-based SSL

[Talukdar et al., EMNLP 2008, 2010]

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Table Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>Set 2</td>
</tr>
<tr>
<td>Bob Dylan</td>
<td>Billy Joel</td>
</tr>
<tr>
<td>Johnny Cash</td>
<td>Johnny Cash</td>
</tr>
<tr>
<td>Billy Joel</td>
<td></td>
</tr>
</tbody>
</table>

Extraction Confidence

Set 1
- Bob Dylan: 0.95
- Johnny Cash: 0.87
- Billy Joel: 0.73

Set 2
- Billy Joel: 0.72
- Johnny Cash: 0.82
- Billy Joel: 0.82
Goal

<table>
<thead>
<tr>
<th>Set 1</th>
<th></th>
<th>Set 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Dylan</td>
<td>0.95</td>
<td>Billy Joel</td>
<td>0.72</td>
</tr>
<tr>
<td>Johnny Cash</td>
<td>0.87</td>
<td>Johnny Cash</td>
<td>0.73</td>
</tr>
<tr>
<td>Billy Joel</td>
<td>0.82</td>
<td>Billy Joel</td>
<td>0.73</td>
</tr>
</tbody>
</table>
Goal

Set 1
- Bob Dylan (0.95)
- Johnny Cash
- Billy Joel

Set 2
- Billy Joel (0.72)
- Johnny Cash (0.73)

Musician

Table Mining
Goal

Set 1
- Bob Dylan
- Johnny Cash
- Billy Joel

Set 2
- Billy Joel
- Johnny Cash

Musician

Set 1
- Bob Dylan: 0.95
- Johnny Cash: 0.87
- Billy Joel: 0.73

Set 2
- Johnny Cash: 0.72
- Billy Joel: 0.82
Goal

Set 1
- Bob Dylan (0.95)
- Johnny Cash
- Billy Joel (0.82)

Set 2
- Billy Joel (0.72)
- Johnny Cash (0.73)

Musician

Table Mining
Goal

Can we infer that **Bob Dylan** is also a musician?

<table>
<thead>
<tr>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Dylan</td>
<td>Billy Joel</td>
</tr>
<tr>
<td>Johnny Cash</td>
<td>Johnny Cash</td>
</tr>
<tr>
<td>Billy Joel</td>
<td></td>
</tr>
</tbody>
</table>

- Set 1: 0.95, 0.87, 0.73
- Set 2: 0.72, 0.82

Musician
Graph Propagation

Set 1

Bob Dylan
0.95
Johnny Cash
0.87
Billy Joel
0.73
0.72

Set 2

Musician, 1.0
Musician, 1.0
Graph Propagation

Set 1

Set 2

Bob Dylan: 0.95
Johnny Cash: 0.87
Billy Joel: 0.73

Seed

Musician, 1.0

Musician, 1.0
Graph Propagation

Set 1

Set 2

Prediction

Seed

Bob Dylan

Johnny Cash

Billy Joel

Musician, 1.0

Musician, 1.0

Musician, 1.0

Musician, 1.0

Musician, 1.0

Musician, 1.0

Seed

0.95

0.87

0.73

0.72

0.82

62
Graph Propagation

Set 1

Set 2

Bob Dylan

Johnny Cash

Billy Joel

Musician, 1.0

Prediction

Seed

62
Graph Propagation

Set 1
- Musician, 0.8
- Set 1

Set 2
- Musician, 1.0
- Set 2

Bob Dylan
- Musician, 1.0

Johnny Cash
- Musician, 1.0

Billy Joel
- Musician, 1.0

Prediction
- 62

Seed
Graph Propagation

Set 1
- Musician, 0.8
- Musician, 1.0
- Seed

Set 2
- Musician, 1.0
- Prediction

Bob Dylan
- Musician, 0.6
- Musician, 1.0

Johnny Cash
- Musician, 1.0

Billy Joel
- Musician, 1.0

Prediction

Seed

62
Graph Propagation

Easily extendible to multiple seeds (classes) for each node
Extraction for Known Instances

Evaluation against WordNet Dataset (38 classes, 8910 instances)

Mean Reciprocal Rank (MRR)

Recall

Patterns
Adsorption
WebTables

924K (class, instance) pairs extracted from 100M web documents
74M (class, instance) pairs extracted from WebTables dataset

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.

Evaluation against WordNet Dataset (38 classes, 8910 instances)

<table>
<thead>
<tr>
<th>Mean Reciprocal Rank (MRR)</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.18</td>
</tr>
<tr>
<td>0.3</td>
<td>0.31</td>
</tr>
<tr>
<td>0.4</td>
<td>0.44</td>
</tr>
<tr>
<td>0.5</td>
<td>0.57</td>
</tr>
<tr>
<td>0.6</td>
<td>0.70</td>
</tr>
</tbody>
</table>

- 924K (class, instance) pairs extracted from 100M web documents
- 74M (class, instance) pairs extracted from WebTables dataset

Patterns
Adsorption
WebTables
# Extracted Pairs

Total classes: 9081

<table>
<thead>
<tr>
<th>Class</th>
<th>Some non-seed Instances found</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFL Players</td>
<td>Tony Gonzales, Thabit Davis, Taylor Stubblefield, Ron Dixon, Rodney Hannan, …</td>
</tr>
</tbody>
</table>
EgoSet [Rong et al., WSDM 2016]
EgoSet [Rong et al., WSDM 2016]

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

\[
\begin{align*}
\text{Barack Obama} & \quad \text{presidentOf} \quad \text{USA} \\
\end{align*}
\]
Two Types of Knowledge Graphs

“Obama was the President of USA.”

![Ontological KG]

- Barack Obama
- USA
- `presidentOf`
+ high precision
Two Types of Knowledge Graphs

“Obama was the President of USA.”

Ontological KG

presidentOf

Barack Obama

USA

+ high precision
+ canonicalized/normalized
Two Types of Knowledge Graphs

“Obama was the President of USA.”

Ontological KG

presidentOf

Barack Obama

USA

+ high precision
+ canonicalized/normalized
- requires supervision
Two Types of Knowledge Graphs

"Obama was the President of USA."

Open KG (Ontology Free)

"Obama"

"was president of"

"USA"

Ontological KG

"Barack Obama"

presidentOf

"USA"

+ high precision
+ canonicalized/normalized
- requires supervision
Two Types of Knowledge Graphs

“Obama was the President of USA.”

Open KG (Ontology Free)

“Obama” \(\rightarrow\) “USA”

+ easy to build, available tools

Ontological KG

“Barack Obama” \(\rightarrow\) “USA”

\text{presidentOf} \rightarrow\text{USA}

+ high precision
+ canonicalized/normalized
- requires supervision
Two Types of Knowledge Graphs

“Obama was the President of USA.”

Open KG (Ontology Free)

“was president of”

“Obama” —> “USA”

+ easy to build, available tools
+ high recall

Ontological KG

presidentOf

Barack Obama —> USA

+ high precision
+ canonicalized/normalized
- requires supervision
Two Types of Knowledge Graphs

“Obama was the President of USA.”

Open KG (Ontology Free)

“was president of”

“Obama”

“USA”

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

Ontological KG

presidentOf

Barack Obama

USA

+ 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
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.”

(Saddam Hussain, invaded, Kuwait)

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

antibiotics (381)
Chlorine (113)
Ozone (61)
Heat (60)
Honey (55)
Benzoyl peroxide (45)

The heat kills the bacteria.
Heat kills the bacteria.
The heat kills bacteria.
Only heat kills bacteria.
Heat kills most bacteria.
Heat can kill the bacteria.
Heat will kill bacteria.
The high heat will kill bacteria.
Heat does kill bacteria.
Open vs. Traditional IE

**Input:**
- Traditional IE: Corpus + Hand-labeled Data
- Open IE: Corpus + Existing resources

**Relations:**
- Traditional IE: Specified in Advance
- Open IE: Discovered Automatically

**Complexity:**
- Traditional IE: $O(D \times R)$, $R$ relations
- Open IE: $O(D)$, $D$ documents

**Output:**
- Traditional IE: relation-specific
- Open IE: 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
ReVerb
Seed Tuples
Training Data
Open Pattern Learning
Bootstrapper
Pattern Templates
WWW
Learning

Extraction
Sentence
Pattern Matching
Tuples
Context Analysis
Ext. Tuples
Enjoy context analysis with this sentence diagram:

John refused to visit Vegas.

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

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

(Venezuela, inflation rate, 96 %)

“Grand Trunk Road is 1,005 kms long.”

(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
Galarraga et al., 2014

• Canonicalize Open KG by clustering synonymous noun 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:**

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

Wooden: 62.3
Entity Linker: 69.1
Galárraga-IDF: 71.5
CESI: 86.3

Average F1 over datasets

60  70  80
Noun Canonicalization

Average precision over datasets

Macro  Micro  Pairwise
Relation Canonicalization

CESI Code: https://github.com/malllabiisc/cesi
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., \textit{undergo}(Patient, Surgery)]
  - starting point in ontological KG construction
  - prepared by experts: expensive and incomplete
Relation Schema Induction

- Relation Schemas [e.g., \textit{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 …”

\[\underbrace{\text{undergo(Patient, Surgery)}}, \text{undergo(Cell, Division)}\]
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

![Plate Diagram of KB-LDA](image)

**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

Datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>KB-LDA</th>
<th>SICTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medline</td>
<td>0.39</td>
<td>0.73</td>
</tr>
<tr>
<td>StackOverflow</td>
<td>0.8</td>
<td>0.94</td>
</tr>
</tbody>
</table>
Binary Schema Induction Results

Datasets
- Medline
- StackOverflow

Accuracy
- KB-LDA
- SICTF

- Medline: 0.39, 0.73
- StackOverflow: 0.8, 0.94

Run Time Comparison
- StackOverflow: 2 hours, 30 hours
- MEDLINE: 1.5 hours, 13 hours

Time in Hours
Binary Schema Induction Results

Datasets: Medline, StackOverflow

- Medline:
  - KB-LDA: 0.39
  - SICTF: 0.73
- StackOverflow:
  - KB-LDA: 0.8
  - SICTF: 0.94

**Relation Schema**

- **StackOverflow**
  - `clicks(A_0, A_1)`:
    - `A_0`: users, client, person
    - `A_1`: link, image, item
  - `refreshes(A_{19}, A_{13})`:
    - `A_{19}`: browser, window, tab
    - `A_{13}`: page, activity, app
  - `can_parse(A_{41}, A_{17})`:
    - `A_{41}`: access, permission, ability
    - `A_{17}`: image file, header file, zip file

**Annotator Judgment**

- Valid
- Invalid

**Run Time Comparison**

- StackOverflow: SICTF (30) vs KB-LDA (2)
- Medline: SICTF (13) vs KB-LDA (1.5)

**SICTF induced schemas**
Binary Schema Induction Results

Datasets:
- Medline
- StackOverflow

<table>
<thead>
<tr>
<th>Relation Schema</th>
<th>Top 3 NPs in Induced Categories</th>
<th>Annotator Judgment</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clicks(A_0, A_1)</code></td>
<td><code>A_0</code>: users, client, person</td>
<td>valid</td>
</tr>
<tr>
<td></td>
<td><code>A_1</code>: link, image, item</td>
<td></td>
</tr>
<tr>
<td><code>refreshes(A_19, A_13)</code></td>
<td><code>A_19</code>: browser, window, tab</td>
<td>valid</td>
</tr>
<tr>
<td></td>
<td><code>A_13</code>: page, activity, app</td>
<td></td>
</tr>
<tr>
<td><code>can_parse(A_41, A_17)</code></td>
<td><code>A_41</code>: access, permission, ability</td>
<td>invalid</td>
</tr>
<tr>
<td></td>
<td><code>A_17</code>: image file, header file, zip file</td>
<td></td>
</tr>
</tbody>
</table>

**SICTF induced schemas**

**MEDLINE**
- `receive(A_1, A_18)`: `A_1`: patient, NUM patients, one patient
  `A_18`: flecaïnide, 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

**Run Time Comparison**
- StackOverflow: SICTF 2 hours, KB-LDA 30 hours
- MEDLINE: SICTF 1.5 hours, KB-LDA 13 hours

SICTF Code: [https://github.com/malllabiisc/sictf](https://github.com/malllabiisc/sictf)
TFBA [Nimishakavi et al., 2018]

- Induces higher-order relation schemas (beyond binary)
- Factorizes higher-order 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-order 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.
TFBA (contd.)

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

<table>
<thead>
<tr>
<th>Relation Schema</th>
<th>NPs from the induced categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shootings</td>
<td></td>
</tr>
</tbody>
</table>
| 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  
                              $B_{16}$: $\mathbf{S} \langle \text{num}\rangle$ million, $\mathbf{S} \langle \text{num}\rangle$, $\mathbf{S} \langle \text{num}\rangle$ billion  
                              $C_3$: $\langle \text{num}\rangle$, year, last season  |
| win($A_2, B_{10}, C_3$) | $A_2$: red sox, team, yankees  
                              $B_{10}$: world series, title, world cup  
                              $C_3$: $\langle \text{num}\rangle$, 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, $\langle \text{num}\rangle$-yard line  |

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