Never Ending Language Learning

Tom Mitchell
Machine Learning Department
Carnegie Mellon University

Thesis:

We’ll never really understand learning until we build machines that
• learn many different things,
• over years,
• and become better learners over time.
NELL: Never-Ending Language Learner

Inputs:
• initial ontology (categories and relations)
• dozen examples of each ontology predicate
• the web
• occasional interaction with human trainers

The task:
• run 24x7, forever
• each day:
  1. extract more facts from the web to populate the ontology
  2. learn to read (perform #1) better than yesterday

NELL today

Running 24x7, since January, 12, 2010

Result:
• KB with > 70 million candidate beliefs, growing daily
• learning to read better each day
• learning to reason, as well as read
• automatically extending its ontology
NELL knowledge fragment

NELL Today

  “BacteriaCausesCondition” “kitchenItem” “ClothingGoesWithClothing” ...

Recently-Learned Facts

<table>
<thead>
<tr>
<th>Instance</th>
<th>Iteration</th>
<th>Date Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>stanford_cardinal is a sports team</td>
<td>809</td>
<td>03-Feb-2014</td>
</tr>
<tr>
<td>unaware_recipes is a recipe</td>
<td>809</td>
<td>03-Feb-2014</td>
</tr>
<tr>
<td>locust_st__ is a street</td>
<td>809</td>
<td>03-Feb-2014</td>
</tr>
<tr>
<td>ty_wigginton is an athlete</td>
<td>809</td>
<td>03-Feb-2014</td>
</tr>
<tr>
<td>tsrc_waco_airport is an airport</td>
<td>809</td>
<td>03-Feb-2014</td>
</tr>
<tr>
<td>roses can represent devotion</td>
<td>810</td>
<td>06-Feb-2014</td>
</tr>
<tr>
<td>jack_nicholson starred in the movie batman</td>
<td>809</td>
<td>03-Feb-2014</td>
</tr>
<tr>
<td>beer is an agricultural product produced in uk</td>
<td>812</td>
<td>15-Feb-2014</td>
</tr>
<tr>
<td>nelson_mandela got married in n1998</td>
<td>814</td>
<td>19-Feb-2014</td>
</tr>
<tr>
<td>wade_boggs is an athlete who injured his/her back</td>
<td>814</td>
<td>19-Feb-2014</td>
</tr>
</tbody>
</table>
How does NELL work?

Semi-Supervised Bootstrap Learning

Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

San Francisco
Berlin
denial

anxiety
selfishness
London

*it’s underconstrained!!*

mayor of arg1
live in arg1

arg1 is home of traits such as arg1
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: \( \sum_{<np,\text{person}> \in \text{labeled data}} |f_1(np) - \text{person}| \)

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

NP:
- **person**
- **f_1(NP)**
- **f_2(NP)**

### Examples
- **is a friend**
- **capitalized?**
- **rung 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]
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
Type 3 Coupling: Learning Relations

Type 3 Coupling: Argument Types

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

over 2500 coupled functions in NELL
Pure EM Approach to Coupled Training

**E**: estimate labels for each function of each unlabeled example

**M**: retrain all functions, using these probabilistic labels

Scaling problem:
- **E** step: 20M NP’s, $10^{14}$ NP pairs to label
- **M** step: 50M text contexts to consider for each function $\rightarrow 10^{10}$ parameters to retrain
- even more URL-HTML contexts…

NELL’s Approximation to EM

**E’** step:
- Consider only a growing subset of the latent variable assignments
  - category variables: up to 250 new NP’s per category per iteration
  - relation variables: add only if confident and args of correct type
  - this set of explicit latent assignments “IS” the knowledge base

**M’** step:
- Each view-based learner re-trains itself from the updated KB
- “context” methods create growing subsets of contexts
**Initial NELL Architecture**

NELL: Learned reading strategies

<table>
<thead>
<tr>
<th>Plays_Sport(arg1,arg2):</th>
<th>Predicate</th>
<th>Feature</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg1_was_playing_arg2</td>
<td>mountain</td>
<td>LAST=peak</td>
<td>1.791</td>
</tr>
<tr>
<td>arg2_megastar_arg1</td>
<td>mountain</td>
<td>LAST=mountain</td>
<td>1.093</td>
</tr>
<tr>
<td>arg2_icons_arg1</td>
<td>mountain</td>
<td>FIRST=mountain</td>
<td>-0.875</td>
</tr>
<tr>
<td>arg1_is_the_tiger_woods_of_arg2</td>
<td>musicArtist</td>
<td>LAST=band</td>
<td>1.853</td>
</tr>
<tr>
<td>arg2_legends_arg1</td>
<td>musicArtist</td>
<td>POS=DT.NNS</td>
<td>1.412</td>
</tr>
<tr>
<td>arg1_announced_arg2</td>
<td>newspaper</td>
<td>LAST=sun</td>
<td>1.330</td>
</tr>
<tr>
<td>arg2_operations_chief_arg1</td>
<td>newspaper</td>
<td>LAST=university</td>
<td>-0.318</td>
</tr>
<tr>
<td>arg2_and_golfing_personals_including_arg1</td>
<td>newspaper</td>
<td>POS=NN_NNS</td>
<td>-0.798</td>
</tr>
<tr>
<td>arg2_player_named_arg1</td>
<td>university</td>
<td>LAST=college</td>
<td>2.076</td>
</tr>
<tr>
<td>arg2_professionals_such_as_arg1</td>
<td>university</td>
<td>PREFIX=uc</td>
<td>1.999</td>
</tr>
<tr>
<td>arg2_icon_arg1</td>
<td>university</td>
<td>LAST=state</td>
<td>1.992</td>
</tr>
<tr>
<td>arg2_great_arg1</td>
<td>university</td>
<td>LAST=university</td>
<td>1.745</td>
</tr>
<tr>
<td>arg2_legends_arg1</td>
<td>university</td>
<td>FIRST=college</td>
<td>-1.381</td>
</tr>
<tr>
<td>arg2_champ_arg1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arg2_great_arg1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arg2_icon_arg1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arg2_greats_like_arg1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If coupled learning is the key, how can we get new coupling constraints?

Key Idea 2:

Discover New Coupling Constraints

- first order, probabilistic horn clause constraints:

\[
\begin{align*}
0.93 \ & \text{athletePlaysSport}(x,y) \leftarrow \text{athletePlaysForTeam}(x,z) \\
& \quad \text{teamPlaysSport}(z,y)
\end{align*}
\]

- learned by data mining the knowledge base
- connect previously uncoupled relation predicates
- infer new unread beliefs
- modified version of FOIL [Quinlan]
Example Learned Horn Clauses

0.95  
athletePlaysSport(?x,basketball) ← athleteInLeague(?x,NBA)

0.93  
athletePlaysSport(?x,?y) ← athletePlaysForTeam(?x,?z),
     teamPlaysSport(?z,?y)

0.91  
teamPlaysInLeague(?x,NHL) ← teamWonTrophy(?x,Stanley_Cup)

0.90  
athleteInLeague(?x,?y) ← athletePlaysForTeam(?x,?z),
     teamPlaysInLeague(?z,?y)

0.88  
cityInState(?x,?y) ← cityCapitalOfState(?x,?y), cityInCountry(?y,USA)

0.62*  
newspaperInCity(?x,New_York) ← companyEconomicSector(?x,media),
       generalizations(?x,blog)

Learned Probabilistic Horn Clause Rules

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

Diagram showing relationships between entities like athlete, coach, sport, team, person, and the associated probabilities.
Inference

If: \( x_1 \) competes with \((x_1, x_2)\) \( x_2 \) economic sector \((x_2, x_3)\) \( x_3 \)

Then: economic sector \((x_1, x_3)\)

Inference by KB Random Walks

KB:

Random walk path type:

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

where \( i \)th feature = probability of arriving at node \( y \) starting at node \( x \), and taking a random walk along path of type \( i \)
Feature = Typed Path
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(\textit{city, 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 CityLocatedInCountry

\textbf{NELL: c:concepts and “noun phrases”}

```
c:penguins  hometown  c:pittsburgh  river flows through  c:monongahela
```

\textit{“Penguins”} \textit{“Pens”} \textit{“Pittsburgh”} \textit{“Pgh”} \textit{“Monongahela”} \textit{“Mon river”}
NELL: c:concepts and "noun phrases"

SVO triples from 500 M dependency parsed web pages (thank you Chris Re!)

- Adding these to graph does not help: too sparse
- BUT, after learning verb phrase classes from latent embedding (NNMF), significantly improves accuracy
  - \{"lies on", "runs through", flows through", \…\}
- Over 15 NELL relations: precision/recall
  - KB only: 0.80 / 0.33
  - KB + SVO_{latent}: 0.87 / 0.42

[Gardner, Talukdar 2013]
Key Idea 3:
Automatically extend ontology

Ontology Extension (1) [Mohamed et al., EMNLP 2011]

Goal:
• Add new relations to ontology

Approach:
• For each pair of categories C1, C2,
  • cluster pairs of known instances, in terms of text contexts that connect them
## Example Discovered Relations

[Mohamed et al. *EMNLP 2011*]

<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 Musician</td>
<td>sitar, George Harrison, tenor sax, Stan Getz, trombone, Tommy Dorsey, Lionel Hampton</td>
<td>ARG1 master ARG2, ARG1 virtuoso ARG2, ARG1 legend ARG2, ARG2 plays ARG1</td>
<td>Master</td>
</tr>
<tr>
<td>Disease Disease</td>
<td>pinched nerve, herniated disk, tennis elbow, tendonitis, blepharospasm, dystonia</td>
<td>ARG1 is due to ARG2, ARG1 is caused by ARG2</td>
<td>IsDueTo</td>
</tr>
<tr>
<td>CellType Chemical</td>
<td>epithelial cells, surfactant neurons, serotonin, mast cells, histamine</td>
<td>ARG1 that release ARG2, ARG2 releasing ARG1</td>
<td>ThatRelease</td>
</tr>
<tr>
<td>Mammals Plant</td>
<td>koala bears, eucalyptus, sheep, grasses, goats, saplings</td>
<td>ARG1 eat ARG2, ARG2 eating ARG1</td>
<td>Eat</td>
</tr>
<tr>
<td>River City</td>
<td>Seine, Paris, Nile, Cairo, Tiber river, Rome</td>
<td>ARG1 in heart of ARG2, ARG1 which flows through ARG2</td>
<td>InHeartOf</td>
</tr>
</tbody>
</table>

## 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
NELL Architecture

Knowledge Base (latent variables)
- Beliefs
- Candidate Beliefs

Evidence Integrator

Text Context patterns (CPL)
Ortographic classifier (CML)
URL specific HTML patterns (SEAL)
Human advice

Actively search for web text (OpenEval)
Infer new beliefs from old (PRA)
Image classifier (NEIL)
Ontology extender (OntExt)

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”
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

What next for NELL?

• micro-reading [Krishnamurthy, Betteridge]
• map each sentence to belief system
  – agree/disagree/accept [Saparov]
• beyond English [Hrushka]
• add computer vision [Gupta, Chen]
• scalable inference over KB [Cohen, Gardner, Talukdar]
• merge with Freebase, Yago [Wijaya, Talukdar]
• goal-driven, targeted reader [Samadi, Kisiel]
• conversational agent on Twitter, Yahoo Answers [Hrushka, Ritter]
thank you

and thanks to:
   Darpa, Google, NSF, Intel, Yahoo!, Microsoft, Fulbright

follow NELL on Twitter: @CMUNELL
browse/download NELL’s KB at http://rtw.ml.cmu.edu