Toward Never-Ending Learning of Semantic Knowledge

Justin Betteridge, Andrew Carlson, Estevam R. Hruschka Jr., Tom M. Mitchell
(with help from Sue Ann Hong, Sophie Wang, Richard Wang)

Carnegie Mellon University

March 2009
Our **Goal**: Never-Ending Language Learning

Goal:

- run 24x7, forever
- each day:
  1. extract more facts from the web to populate and extend initial ontology
  2. learn to read better than yesterday
Our **Goal**: Never-Ending Language Learning

Goal:
- run 24x7, forever
- each day:
  1. extract more facts from the web to populate initial ontology
  2. learn to read better than yesterday

Today…

Given:
- initial ontology defining dozens of classes and relations
- 10-20 seed examples of each

Task:
- learn to extract / extract to learn
- running over 200M web pages, for a few days
Browse the KB

- ~ 18,000+ entities, ~ 30,000 extracted beliefs
- learned from 10-20 seed examples, 200M unlabeled web pages
- ~ 2 days computation on M45 cluster (thanks Yahoo!)

Initial ontology: Initial ontology
learned KB: learned KB

or get it from the web:
http://rtw.ml.cmu.edu/kb/RTW_KB_2009_03_19_ORS/
The Problem with Semi-Supervised Bootstrap Learning

Paris
Pittsburgh
Seattle
Cupertino

San Francisco
Austin
denial

mayor of arg1
live in arg1

arg1 is home of
traits such as arg1

it's underconstrained!!
The Key to Accurate Semi-Supervised Learning

**hard** (underconstrained) semi-supervised learning problem

**much easier** (more constrained) semi-supervised learning problem

Krzyzewski coaches the Blue Devils.
Constraining semi-supervised learning 1

Wish to learn $f : X \rightarrow Y$

  e.g., $\text{city} : \text{NounPhraseInSentence} \rightarrow \{0,1\}$

Constraint type 1 (co-training):
  if $X$ can be split into redundantly sufficient $X_1, X_2$
  then learn both $f_1 : X_1 \rightarrow Y$, and $f_2 : X_2 \rightarrow Y$

Constraining semi-supervised learning 2

Wish to learn $f: X \rightarrow Y$

- e.g., city: NounPhraseInSentence $\rightarrow \{0, 1\}$

Constraint type 2: couple training of multiple classes
Ontology provides coupling constraints

Luke is mayor of Pittsburgh.
Constraining semi-supervised learning 2

Wish to learn $f: X \rightarrow Y$

e.g., city: NounPhraseInSentence $\rightarrow \{0,1\}$

Constraint type 2: couple training of multiple classes
Ontology provides coupling constraints

Luke is mayor of Pittsburgh.
Constraining semi-supervised learning 3

Constraint type 3 (couple training of classes and relations)

Luke is mayor of Pittsburgh.

\[ \text{mayorOf}(X1, X2)? \]

location? city? politician?

\[ X1 \]

\[ X2 \]
Coupled Bootstrap Learner algorithm

**Algorithm 1: CBL Algorithm**

<table>
<thead>
<tr>
<th>Input:</th>
<th>An ontology $\mathcal{O}$, and text corpus $C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>Trusted instances/patterns for each predicate</td>
</tr>
</tbody>
</table>

Share initial instances/patterns among predicates;

for $i = 1, 2, \ldots, \infty$ do

foreach predicate $p \in \mathcal{O}$ do

- **Extract** new candidate instances/patterns;
- **Filter** candidates;
- **Train** instance/pattern classifiers;
- **Assess** candidates using trained classifiers;
- **Promote** highest-confidence candidates;

end

Share promoted items among predicates;

end

In the **ontology**: categories, relations, seed instances and patterns, type information, sharing enforces mutual exclusion, subset relations, and type checking.

**Extraction (M45)**: Arg1 HQ in Arg2 $\rightarrow$ (CBC || Toronto), (Adobe || San Jose), …

**Filtering (M45)**: CBC || Toronto $\rightarrow$ Not enough evidence arg1 of arg2 $\rightarrow$ too general arg2 is headquarters for chipmaker arg1 $\rightarrow$ too specific

**Assessment (M45)**: Classify candidate instances with a Naïve Bayes classifier. Features related to strength of occurrence with each pattern.

Promote top ranked instances and patterns. Use type-checking and board.

Score patterns with estimate.
learned extraction patterns: Company

retailers_like__ such_clients_as__ an_operating_business_of__ being_acquired_by__
firms_such_as__ a_flight_attendant_for__ chains_such_as__
industry_leaders_such_as__ advertisers_like__ social_networking_sites_such_as__
a_senior_manager_at__ competitors_like__ stores_like__ is_an_ebay_company
discounters_like__ a_distribution_deal_with__ popular_sites_like__
a_company_such_as__ vendors_such_as__ rivals_such_as__ competitors_such_as__
has Been quoted in the__ providers_such_as__ company_research_for__
providers_like__ giants_such_as__ a_social_network_like__ popular_websites_like__
multinationals_like__ social_networks_such_as__ the_former_ceo_of__
a_software_engineer_at__ a_store_like__ video_sites_like__
a_social_networking_site_like__ giants_like__ a_company_like__
premieres on__ corporations_such_as__ corporations_like__ professional_profile_on__ outlets_like__
the_executives_at__ stores_such_as__ is the only carrier a_big_company_like__
social_media_sites_such_as__ has an article today manufacturers_such_as__
companies_like__ social_media_sites_like__ companies__including__ firms_like__
networking_websites_such_as__ networks_like__ carriers_like__
social_networking_websites_like__ an_executive_at__ insured_via__
provides_dialup_access a_patent_infringement_lawsuit_against__
social_networking_sites_like__ social_network_sites_like__ carriers_such_as__
are_shipped_via__ social_sites_like__ a_licensing_deal_with__ portals_like__
vendors_like__ the_accounting_firm_of__ industry_leaders_like__ retailers_such_as__
chains_like__ prior_fiscal_years_for__ such_firms_as__ provided_free_by__
manufacturers_like__ airlines_like__ airlines_such_as__
learned extraction patterns: playsSport(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_legends_like_arg1  arg2_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_legends_are_steffi_graf_and_arg1  arg2_great_arg1  arg2_champ_arg1  arg2_legends_such_as_arg1  arg2_greats_such_as_arg1  arg2_professionals_such_as_arg1  arg2_course_designed_by_arg1  arg2_hit_by_arg1  arg2_course_architects_including_arg1  arg2_greats_arg1  arg2_icon_arg1  arg2_legends_as_arg1  arg1_retires_from_arg2  arg2_phenom_arg1  arg2_course_architects_robert_trent_jones_and_arg1  arg2_sensation_arg1  arg2_course_architects_like_arg1  arg2_legends_such_as_arg1  arg2_legends_arndt_palmer_and_arg1  arg2_legends_such_as_arg1  arg2_hit_by_arg1  arg2_legends_such_as_arg1  arg2_player_is_arg1  arg2_serena_williams_and_arg1  arg2_legends_such_as_arg1  arg2_legends_like_arg1  arg2_player_was_arg1  arg2_god_arg1  arg2_idol_arg1  arg1_was_born_to_play_arg2  arg2_great_arg1  arg2_champ_arg1  arg2_legends_like_arg1  arg2_player_is_arg1  arg2_legends_like_arg1  arg2_legends_as_arg1  arg2_autographed_by_arg1  arg2_related_quotations_spoken_by_arg1  arg2_courses_were_designed_by_arg1  arg1_has_retired_from_arg2  arg2_legends_including_arg1  arg1_can_hit_a_arg2  arg2_legends_like_arg1  arg2_courses_designed_by_legends_arg1  arg2_course_was_designed_by_arg1  arg2_champion_arg1  arg2_legends_of_all_time_is_arg1  arg2_fan_knows_arg1  arg1_learned_to_play_arg2  arg1_is_the_best_player_in_arg2  arg2_signed_by_arg1  arg2_champion_arg1
Experimental Evaluation

- 31 predicates
  - 15 relations, 16 categories
- Domains:
  - Companies
  - Sports
- Run for 15 iterations:
  - Full system
  - No Sharing of promoted items
  - No Relation/Category coupling
- Evaluated a sample of promoted items
<table>
<thead>
<tr>
<th>Predicate</th>
<th>5 iterations</th>
<th></th>
<th></th>
<th></th>
<th>10 iterations</th>
<th></th>
<th></th>
<th></th>
<th>15 iterations</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Full</td>
<td>NS</td>
<td>NCR</td>
<td></td>
<td>Full</td>
<td>NS</td>
<td>NCR</td>
<td></td>
<td>Full</td>
<td>NS</td>
<td>NCR</td>
</tr>
<tr>
<td>Actor</td>
<td></td>
<td>93</td>
<td>100</td>
<td>100</td>
<td></td>
<td>93</td>
<td>97</td>
<td>100</td>
<td></td>
<td>100</td>
<td>97</td>
<td>100</td>
</tr>
<tr>
<td>Athlete</td>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td>100</td>
<td>93</td>
<td>100</td>
<td></td>
<td>100</td>
<td>73</td>
<td>100</td>
</tr>
<tr>
<td>Board Game</td>
<td></td>
<td>93</td>
<td>76</td>
<td>93</td>
<td></td>
<td>89</td>
<td>27</td>
<td>93</td>
<td></td>
<td>89</td>
<td>30</td>
<td>93</td>
</tr>
<tr>
<td>City</td>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td>100</td>
<td>97</td>
<td>100</td>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Coach</td>
<td></td>
<td>100</td>
<td>63</td>
<td>73</td>
<td></td>
<td>97</td>
<td>53</td>
<td>43</td>
<td></td>
<td>97</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>Company</td>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td>97</td>
<td>90</td>
<td>97</td>
<td></td>
<td>100</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>Country</td>
<td></td>
<td>60</td>
<td>40</td>
<td>60</td>
<td></td>
<td>30</td>
<td>43</td>
<td>27</td>
<td></td>
<td>40</td>
<td>23</td>
<td>40</td>
</tr>
<tr>
<td>Economic Sector</td>
<td></td>
<td>77</td>
<td>63</td>
<td>73</td>
<td></td>
<td>57</td>
<td>67</td>
<td>67</td>
<td></td>
<td>50</td>
<td>63</td>
<td>40</td>
</tr>
<tr>
<td>Hobby</td>
<td></td>
<td>67</td>
<td>63</td>
<td>67</td>
<td></td>
<td>40</td>
<td>40</td>
<td>57</td>
<td></td>
<td>20</td>
<td>23</td>
<td>30</td>
</tr>
<tr>
<td>Person</td>
<td></td>
<td>97</td>
<td>97</td>
<td>90</td>
<td></td>
<td>97</td>
<td>93</td>
<td>97</td>
<td></td>
<td>97</td>
<td>97</td>
<td>93</td>
</tr>
<tr>
<td>Politician</td>
<td></td>
<td>93</td>
<td>93</td>
<td>97</td>
<td></td>
<td>73</td>
<td>53</td>
<td>90</td>
<td></td>
<td>90</td>
<td>53</td>
<td>87</td>
</tr>
<tr>
<td>Product</td>
<td></td>
<td>97</td>
<td>87</td>
<td>90</td>
<td></td>
<td>90</td>
<td>87</td>
<td>100</td>
<td></td>
<td>97</td>
<td>90</td>
<td>77</td>
</tr>
<tr>
<td>Product Type</td>
<td></td>
<td>93</td>
<td>93</td>
<td>90</td>
<td></td>
<td>70</td>
<td>73</td>
<td>97</td>
<td></td>
<td>77</td>
<td>80</td>
<td>67</td>
</tr>
<tr>
<td>Scientist</td>
<td></td>
<td>100</td>
<td>90</td>
<td>97</td>
<td></td>
<td>97</td>
<td>63</td>
<td>97</td>
<td></td>
<td>93</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>Sport</td>
<td></td>
<td>100</td>
<td>90</td>
<td>100</td>
<td></td>
<td>93</td>
<td>67</td>
<td>83</td>
<td></td>
<td>97</td>
<td>27</td>
<td>90</td>
</tr>
<tr>
<td>Sports Team</td>
<td></td>
<td>100</td>
<td>97</td>
<td>100</td>
<td></td>
<td>97</td>
<td>70</td>
<td>100</td>
<td></td>
<td>90</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>Category Average</td>
<td></td>
<td>92</td>
<td>84</td>
<td>89</td>
<td></td>
<td>82</td>
<td>70</td>
<td>84</td>
<td></td>
<td>83</td>
<td>63</td>
<td>79</td>
</tr>
<tr>
<td>Acquired(Company, Company)</td>
<td></td>
<td>77</td>
<td>77</td>
<td>80</td>
<td></td>
<td>67</td>
<td>80</td>
<td>47</td>
<td></td>
<td>70</td>
<td>63</td>
<td>47</td>
</tr>
<tr>
<td>CEOOf(Person, Company)</td>
<td></td>
<td>97</td>
<td>87</td>
<td>100</td>
<td></td>
<td>90</td>
<td>87</td>
<td>97</td>
<td></td>
<td>90</td>
<td>80</td>
<td>83</td>
</tr>
<tr>
<td>CoachesTeam(Coach, Sports Team)</td>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td>100</td>
<td>100</td>
<td>97</td>
<td></td>
<td>100</td>
<td>100</td>
<td>90</td>
</tr>
<tr>
<td>CompetesIn(Company, Econ. Sector)</td>
<td></td>
<td>97</td>
<td>97</td>
<td>80</td>
<td></td>
<td>100</td>
<td>93</td>
<td>67</td>
<td></td>
<td>97</td>
<td>63</td>
<td>60</td>
</tr>
<tr>
<td>CompetesWith(Company, Company)</td>
<td></td>
<td>93</td>
<td>80</td>
<td>60</td>
<td></td>
<td>77</td>
<td>70</td>
<td>37</td>
<td></td>
<td>70</td>
<td>60</td>
<td>43</td>
</tr>
<tr>
<td>HasOfficesIn(Company, City)</td>
<td></td>
<td>97</td>
<td>93</td>
<td>40</td>
<td></td>
<td>93</td>
<td>90</td>
<td>27</td>
<td></td>
<td>93</td>
<td>57</td>
<td>30</td>
</tr>
<tr>
<td>HasOperationsIn(Company, Country)</td>
<td></td>
<td>100</td>
<td>95</td>
<td>50</td>
<td></td>
<td>100</td>
<td>97</td>
<td>40</td>
<td></td>
<td>90</td>
<td>83</td>
<td>13</td>
</tr>
<tr>
<td>HeadquarteredIn(Company, City)</td>
<td></td>
<td>77</td>
<td>90</td>
<td>20</td>
<td></td>
<td>70</td>
<td>77</td>
<td>27</td>
<td></td>
<td>70</td>
<td>60</td>
<td>7</td>
</tr>
<tr>
<td>LocatedIn(City, Country)</td>
<td></td>
<td>90</td>
<td>67</td>
<td>57</td>
<td></td>
<td>63</td>
<td>50</td>
<td>43</td>
<td></td>
<td>73</td>
<td>50</td>
<td>30</td>
</tr>
<tr>
<td>PlaysFor(Athlete, Sports Team)</td>
<td></td>
<td>100</td>
<td>100</td>
<td>0</td>
<td></td>
<td>100</td>
<td>97</td>
<td>7</td>
<td></td>
<td>100</td>
<td>43</td>
<td>0</td>
</tr>
<tr>
<td>PlaysSport(Athlete, Sport)</td>
<td></td>
<td>100</td>
<td>100</td>
<td>27</td>
<td></td>
<td>93</td>
<td>80</td>
<td>10</td>
<td></td>
<td>100</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>TeamPlaysSport(Sports Team, Sport)</td>
<td></td>
<td>100</td>
<td>100</td>
<td>77</td>
<td></td>
<td>100</td>
<td>97</td>
<td>80</td>
<td></td>
<td>93</td>
<td>83</td>
<td>67</td>
</tr>
<tr>
<td>Produces(Company, Product)</td>
<td></td>
<td>91</td>
<td>83</td>
<td>90</td>
<td></td>
<td>83</td>
<td>93</td>
<td>67</td>
<td></td>
<td>93</td>
<td>80</td>
<td>57</td>
</tr>
<tr>
<td>HasType(Product, Product Type)</td>
<td></td>
<td>73</td>
<td>63</td>
<td>17</td>
<td></td>
<td>33</td>
<td>67</td>
<td>33</td>
<td></td>
<td>40</td>
<td>57</td>
<td>27</td>
</tr>
<tr>
<td>Relation Average</td>
<td></td>
<td>92</td>
<td>88</td>
<td>57</td>
<td></td>
<td>84</td>
<td>84</td>
<td>48</td>
<td></td>
<td>84</td>
<td>66</td>
<td>42</td>
</tr>
<tr>
<td>All</td>
<td></td>
<td>92</td>
<td>86</td>
<td>74</td>
<td></td>
<td>83</td>
<td>76</td>
<td>68</td>
<td></td>
<td>84</td>
<td>64</td>
<td>62</td>
</tr>
</tbody>
</table>

Table 1: Precision (%) for each predicate. Results are presented after 5, 10, and 15 iterations, for the Full, No Sharing (NS), and No Category/Relation Coupling (NCR) configurations of CBL.
Extending Freebase

<table>
<thead>
<tr>
<th>Category</th>
<th>Freebase Matches</th>
<th>CBL Instances</th>
<th>Est. Prec.</th>
<th>Est. New Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>465</td>
<td>522</td>
<td>100</td>
<td>57</td>
</tr>
<tr>
<td>Athlete</td>
<td>54</td>
<td>117</td>
<td>100</td>
<td>63</td>
</tr>
<tr>
<td>Board Game</td>
<td>6</td>
<td>18</td>
<td>89</td>
<td>10</td>
</tr>
<tr>
<td>City</td>
<td>1665</td>
<td>1799</td>
<td>100</td>
<td>134</td>
</tr>
<tr>
<td>Company</td>
<td>995</td>
<td>1937</td>
<td>100</td>
<td>942</td>
</tr>
<tr>
<td>Econ. Sector</td>
<td>137</td>
<td>1541</td>
<td>50</td>
<td>634</td>
</tr>
<tr>
<td>Politician</td>
<td>74</td>
<td>962</td>
<td>90</td>
<td>792</td>
</tr>
<tr>
<td>Product</td>
<td>0</td>
<td>1259</td>
<td>97</td>
<td>1221</td>
</tr>
<tr>
<td>Sports Team</td>
<td>139</td>
<td>414</td>
<td>90</td>
<td>234</td>
</tr>
<tr>
<td>Sport</td>
<td>134</td>
<td>613</td>
<td>97</td>
<td>461</td>
</tr>
</tbody>
</table>

Table 3: Estimated numbers of ‘New Instances’, which are correct instances promoted by CBL in the Full 15 iteration run which do not have a match in Freebase, and the values used in calculating them (number of Freebase/CBL matches, number of CBL instances, and the estimated precision of CBL for the predicate).
If the key to accurate self-supervised learning is coupling the training of many functions, then how can we create even more coupling?

1. incorporate additional learners whose errors will be independent of current learners (e.g., based on HTML)

Krzyzewski coaches the Blue Devils.
SEAL

Set Expander for Any Language

SEAL

For each class being learned,
   On each iteration of CBL
      Train SEAL on examples extracted by CBL, then apply
      Allow SEAL to suggest additional examples

Experiment:
  15 classes, ~15000 examples extracted by CBL
  result: ~5000 examples suggested by SEAL (includes duplicates)

Typical learned company extractor:
  at http://www.transnationale.org/countries/panp.php
  pattern: "</a>, <A HREF="../companies/?X.php">"
  extracts: banco brasil
If the key to accurate self-supervised learning is coupling the training of many functions, then how can we create even more coupling?

2. allow learner to discover new coupling constraints (by datamining the extracted beliefs)
Learned Probabilistic Horn Clause Rules

- 40 learned rules for teamPlaySport, playSport,
- when applied, inferred 124 new beliefs
  - e.g., teamPlaysSport(Caps,hockey),
  - playSport(JasonGiambi,baseball)

\[
\begin{align*}
0.84 \text{playsSport}(?x,?y) & \leftarrow \text{playsFor}(?x,?z), \text{teamPlaysSport}(?z,?y) \\
0.70 \text{playSport}(?x,\text{baseball}) & \leftarrow \text{playsFor}(?x,\text{cubs}) \\
\ldots \\
0.81 \text{teamPlaysSport}(?x,?y) & \leftarrow \text{playsForTeam}(?x,?z), \text{playSport}(?z,?y) \\
0.70 \text{teamPlaysSport}(?x,\text{basketball}) & \leftarrow \text{playsAgainst}(?x,\text{pistons}) \\
0.64 \text{teamPlaysSport}(?x,?y) & \leftarrow \text{playsAgainst}(?x,?z), \text{teamPlaysSport}(?z,?y) \\
\ldots
\end{align*}
\]
Learned Probabilistic Horn Clause Rules

$$0.81 \quad \text{teamPlaysSport}(\texttt{?x,?y}) \leftarrow \text{playsForTeam}(\texttt{?x,?z}), \text{playSport}(\texttt{?z,?y})$$
Summary

For never-ending language learning, the key is achieving accurate semi-supervised training

→ Constrain learning by coupling the training of many types of knowledge (functions)
  – sample complexity decreases as ontology size increases

→ Want an architecture in which current learning makes future learning even more accurate
  -- learn symbolic rules which become new probabilistic constraints

→ Want architecture where self-consistency ≈ correctness