Reading the Web: Advanced Statistical Language Processing

www.cs.cmu.edu/~tom/rtw09/

Machine Learning 10-709

September 10, 2009

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University
The Plan

• What will you get out of this class?
  – knowledge of state of the art in semi-supervised learning, statistical NLP, never-ending-learning
  – an opportunity to advance it
  – a research infrastructure you might use in the future

• What will you be expected to do?
  – read, discuss, critique research papers
  – design and perform a research project in this area

• What will we build on?
  – the RTW project data and knowledge base
The Goals*

• Build the first cumulative never-ending learner

• Advance state of Natural Language Understanding

• Build and publish the world’s largest structured knowledge base

* choose research problems wisely – 2/3 of success in research is (re)choosing the problem
The Problem Specification

Inputs:
• initial ontology
• handful of examples of each predicate in ontology
• the web
• occasional access to human trainer

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

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 per predicate, 200M unlabeled web pages
• ~ 2 days computation on M45 cluster

Initial ontology: Initial ontology

After a few days of self-supervised learning:
http://rtw.ml.cmu.edu/sslInlp09/index.html
http://rtw.ml.cmu.edu/wsdm10_online/
Semi-Supervised Bootstrap Learning

Paris
Pittsburgh
Seattle
Cupertino

San Francisco
Austin
denial

anxiety
selfishness
Berlin

it’s underconstrained!!

mayor of arg1
live in arg1

arg1 is home of traits such as arg1
The Key to Accurate Semi-Supervised Learning

Krzyzewski coaches the Blue Devils.

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

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

The Key: Couple the training of many functions to make unlabeled data more informative
Coupled training type 1

Wish to learn $f : X \rightarrow Y$

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

Coupling type 1 (co-training): Learn 2 functions with different input features

$f_1 : X_1 \rightarrow Y$, and $f_2 : X_2 \rightarrow Y$

Coupling: force their outputs to agree over unlabeled examples


*co-training [Blum, Mitchell 1998]
Coupled training type 2

Wish to learn $f: X \rightarrow Y_1$, $f: X \rightarrow Y_2$, where $g(y_1, y_2)$

city: NounPhraseInSentence $\rightarrow \{0, 1\}$

politician: NounPhraseInSentence $\rightarrow \{0, 1\}$

Constraint type 2: force outputs to satisfy $g(y_1, y_2)$

Ontology provides coupling constraints

Luke is mayor of Pittsburgh.
Coupled training type 3

Constraint type 3 (argument type consistency)

\[ \text{mayorOf: NP1} \times \text{NP2} \rightarrow \{0,1\} \]
\[ \text{city: NP1} \rightarrow \{0,1\} \]
\[ \text{politician: NP2} \rightarrow \{0,1\} \]

Luke is mayor of Pittsburgh.
Coupled Bootstrap Learner algorithm

In the ontology: categories, relations, seed instances and patterns, type information, mutual exclusion and subset relations

Sharing enforces mutual exclusion, subset relations, and type checking

Extraction (M45): Arg1 HQ in Arg2 → (CBC || Toronto), (Adobe || San Jose), …
Micron || Boise → arg2 is headquarters for chipmaker arg1

Filtering (M45): CBC || Toronto → Not enough evidence
ARG1 of ARG2 → arg1 arg2

Assessment (M45): Classify candidate instances with a Naïve Bayes classifier
Score patterns with estimate of precision
Promote top ranked instances and patterns. Use type-checking

Score patterns with estimate of precision

Algorithm 1: CBL Algorithm

**Input:** An ontology \( \mathcal{O} \), and text corpus \( C \)

**Output:** Trusted instances/patterns for each predicate

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
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__ premiers 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_players_are_steffi_graf_and_arg1  arg2_great_arg1
arg2_champ_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_stars_like_arg1  arg2_pros_like_arg1
arg1_retires_from_arg2  arg2_phenom_arg1  arg2_lesson_from_arg1
arg2_architects_robert_trent_jones_and_arg1  arg2_sensation_arg1  arg2_architects_like_arg1
arg2_pros_arg1  arg2_stars_venus_and_arg1  arg2_legends_arnold_palmer_and_arg1
arg2_hall_of_famer_arg1  arg2_racket_in_arg1  arg2_superstar_arg1  arg2_legend_arg1
arg2_legends_such_as_arg1  arg2_players_is_arg1  arg2_pro_arg1  arg2_player_was_arg1
arg2_god_arg1  arg2_idol_arg1  arg1_was_born_to_play_arg2  arg2_star_arg1
arg2_hero_arg1  arg2_course_architect_arg1  arg2_players_are_arg1
arg1_retired_from_professional_arg2  arg2_legends_as_arg1  arg2_autographed_by_arg1
arg2_related_quotations_spoken_by_arg1  arg2_courses_were_designed_by_arg1
arg2_player_since_arg1  arg2_match_between_arg1  arg2_course_was_designed_by_arg1
arg1_has_retired_from_arg2  arg2_player_arg1  arg1_can_hit_a_arg2
arg2_legends_including_arg1  arg2_player_than_arg1  arg2_legends_like_arg1
arg2_courses_designed_by_legends_arg1  arg2_player_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
If the key to accurate self-supervised learning is coupling the training of many functions, then how can we create even more coupling?

1. introduce additional coupling by adding a learner that uses HTML features instead of free text features.

Krzyzewski coaches the Blue Devils.
SEAL
Set Expander for Any Language


Seeds

- ford
- toyota
- nissan

Extraction

- ford
- toyota
- nissan
- honda
SEAL

For each class being learned,

On each iteration

Retrain CBL from current KB, allow it to add to KB
Retrain SEAL from current KB, allow it to add to KB

Typical learned SEAL extractors:

<table>
<thead>
<tr>
<th>URL</th>
<th>Wrapper</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.shopcarparts.com/">http://www.shopcarparts.com/</a></td>
<td>.html&quot; CLASS=&quot;shopcp&quot;&gt;[... ] Parts&lt;/A&gt; &lt;br&gt;</td>
<td>acura, audi, bmw, buick, cadillac, chevrolet, chevy, chrysler, daewoo, daihatsu, dodge, ea,</td>
</tr>
<tr>
<td><a href="http://www.allautoreviews.com/">http://www.allautoreviews.com/</a></td>
<td>&lt;/a&gt;&lt;br&gt; &lt;a href=&quot;auto_reviews/[...]/</td>
<td>acura, audi, bmw, buick, cadillac, chevrolet, chrysler, dodge, ford, gmc, honda, hyundai, i,</td>
</tr>
<tr>
<td><a href="http://www.hertrichs.com/">http://www.hertrichs.com/</a></td>
<td>&lt;li class=&quot;franchise [...]&quot;&gt; &lt;h4&gt;&lt;a href=&quot;#&quot;</td>
<td>buick, chevrolet, chrysler, dodge, ford, gmc, isuzu, jeep, lincoln, mazda, mercury, nissan,</td>
</tr>
<tr>
<td>Predicate</td>
<td>Precision (%)</td>
<td>Promoted Instances (#)</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>AcademicField</td>
<td>70 83 90 97</td>
<td>46 903 203 1000</td>
</tr>
<tr>
<td>Actor</td>
<td>100 32 100 97</td>
<td>190 1000 1000 1000</td>
</tr>
<tr>
<td>Animal</td>
<td>80 50 90 70</td>
<td>741 1000 144 974</td>
</tr>
<tr>
<td>Athlete</td>
<td>87 17 100 87</td>
<td>132 930 276 1000</td>
</tr>
<tr>
<td>AwardTrophyTournament</td>
<td>57 7 53 7</td>
<td>86 902 146 1000</td>
</tr>
<tr>
<td>BoardGame</td>
<td>80 13 70 77 90</td>
<td>10 907 126 1000</td>
</tr>
<tr>
<td>BodyPart</td>
<td>77 17 97 63 93</td>
<td>176 922 80 1000</td>
</tr>
<tr>
<td>Building</td>
<td>33 50 30 0 93</td>
<td>597 1000 57 1000</td>
</tr>
<tr>
<td>Celebrity</td>
<td>100 90 100 100</td>
<td>347 1000 72 747</td>
</tr>
<tr>
<td>CEO</td>
<td>33 30 100 77</td>
<td>3 902 322 1000</td>
</tr>
<tr>
<td>City</td>
<td>97 100 97 87 97</td>
<td>1000 1000 368 1000</td>
</tr>
<tr>
<td>Clothing</td>
<td>97 20 43 27 97</td>
<td>83 973 167 1000</td>
</tr>
<tr>
<td>Coach</td>
<td>93 63 100 86 100</td>
<td>1000 1000 245 1000</td>
</tr>
<tr>
<td>Company</td>
<td>97 83 100 100 97</td>
<td>1000 1000 245 1000</td>
</tr>
<tr>
<td>Conference</td>
<td>93 53 97 90 100</td>
<td>95 990 437 928</td>
</tr>
<tr>
<td>Country</td>
<td>57 33 97 37 93</td>
<td>1000 1000 130 1000</td>
</tr>
<tr>
<td>EconomicSector</td>
<td>60 23 100 10 77</td>
<td>1000 1000 34 1000</td>
</tr>
<tr>
<td>Emotion</td>
<td>77 53 87 60 83</td>
<td>483 992 183 1000</td>
</tr>
<tr>
<td>Food</td>
<td>90 70 97 80 100</td>
<td>811 1000 89 1000</td>
</tr>
<tr>
<td>Furniture</td>
<td>100 0 57 57 90</td>
<td>55 963 215 1000</td>
</tr>
<tr>
<td>Hobby</td>
<td>77 33 77 50 90</td>
<td>357 930 77 1000</td>
</tr>
<tr>
<td>KitchenItem</td>
<td>73 3 88 13 100</td>
<td>11 900 8 960</td>
</tr>
<tr>
<td>Mammal</td>
<td>83 50 93 50 90</td>
<td>224 1000 154 1000</td>
</tr>
<tr>
<td>Movie</td>
<td>97 57 97 100 100</td>
<td>718 1000 566 1000</td>
</tr>
<tr>
<td>NewspaperCompany</td>
<td>90 60 60 97 100</td>
<td>1000 1000 600 1000</td>
</tr>
<tr>
<td>Politician</td>
<td>80 60 97 37 100</td>
<td>178 990 30 1000</td>
</tr>
<tr>
<td>Product</td>
<td>90 83 - 77 70 1000 1000 0 999</td>
<td>127</td>
</tr>
<tr>
<td>ProductType</td>
<td>73 63 27 63 50</td>
<td>712 1000 31 1000</td>
</tr>
<tr>
<td>Profession</td>
<td>73 53 - 57 93</td>
<td>916 973 0 1000</td>
</tr>
<tr>
<td>ProfessionalOrganization</td>
<td>93 63 100 77 87</td>
<td>104 943 58 1000</td>
</tr>
<tr>
<td>Reptile</td>
<td>95 3 90 27 100</td>
<td>19 912 149 1000</td>
</tr>
<tr>
<td>Room</td>
<td>64 0 33 7 100</td>
<td>25 913 12 643</td>
</tr>
<tr>
<td>Scientist</td>
<td>97 30 100 17 100</td>
<td>83 971 928 1000</td>
</tr>
<tr>
<td>Shape</td>
<td>77 7 7 7 85</td>
<td>43 985 28 733</td>
</tr>
<tr>
<td>Sport</td>
<td>77 13 63 63 83 73</td>
<td>283 1000 225 1000</td>
</tr>
<tr>
<td>SportsEquipment</td>
<td>20 10 57 23 23</td>
<td>58 902 52 1000</td>
</tr>
<tr>
<td>SportsLeague</td>
<td>100 7 90 27 86</td>
<td>41 901 10 1000</td>
</tr>
<tr>
<td>SportsTeam</td>
<td>90 30 87 87 87</td>
<td>301 903 864 944</td>
</tr>
<tr>
<td>Stadium</td>
<td>93 57 53 63 90</td>
<td>102 767 944 1000</td>
</tr>
<tr>
<td>StateOrProvince</td>
<td>77 63 83 93 77</td>
<td>202 1000 114 1000</td>
</tr>
<tr>
<td>Tool</td>
<td>40 13 93 90 97</td>
<td>561 1000 713 1000</td>
</tr>
<tr>
<td>Trait</td>
<td>53 40 52 47 97</td>
<td>234 1000 21 1000</td>
</tr>
<tr>
<td>University</td>
<td>93 97 100 90 93</td>
<td>1000 1000 961 1000</td>
</tr>
<tr>
<td>Vehicle</td>
<td>67 30 50 13 77</td>
<td>460 1000 50 1000</td>
</tr>
<tr>
<td>Average</td>
<td>78 41 78 59 90</td>
<td>360 960 271 976</td>
</tr>
<tr>
<td>Weighted average</td>
<td>79 42 86 59 91</td>
<td>360 960 271 976</td>
</tr>
</tbody>
</table>

Table 2: Precision (%) and counts of promoted instances for each category using CPL, UPL, CSEAL, SEAL, MBL.
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 mining its extracted beliefs)

Krzyzewski coaches the Blue Devils.
Learned Probabilistic Horn Clause Rules

- 40 learned rules for teamPlaySport, playSport,
- when applied, inferred over 800 new beliefs
  - e.g., teamPlaysSport(Caps,hockey),
  - playSport(JasonGiambi,baseball)

0.84 \text{playsSport}(?x,?y) \leftarrow \text{playsFor}(?x,?z), \text{teamPlaysSport}(?z,?y)
0.70 \text{playsSport}(?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
Learned Probabilistic Horn Clause Rules

0.81 teamPlaysSport(?x,?y) ← playsForTeam(?x,?z), playSport(?z,?y)
Summary: what we have to work from

Data/Knowledge:
• KB with $10^4$ extracted beliefs, .85-.90 accurate
• statistics on co-occurrence frequency of
  – $10^5$ NPs x $10^5$ contexts

Learning algorithms
• coupled semi-supervised learning of name entity extractors, relation extractors
• learning probabilistic first-order horn clauses
Homework 1: due next Thursday

- get co-occurrence data: $10^5$ NP’s x $10^5$ Contexts
- look at [http://rtw.ml.cmu.edu/wsdm10_online/](http://rtw.ml.cmu.edu/wsdm10_online/)
  - labeled data: “Instances Promoted by Meta-Bootstrap Learner”
- do something interesting, prepare a 2 slide, 3 min presentation

Example:
learn to classify <NP,Context>, or just NP’s as city, person, emotion, ...
  - supervised, semi-supervised, unsupervised, ...
Projects Ideas 1

- Add a morphology-based entity extractor
  - Omalinski is probably a person
  - yet another redundant information source

- Ultra High-dimensional training
  - each noun phrase as bag of $10^6$ contexts
  - each context as bag of $10^5$ noun phrases

- Add first self-reflection capability
  - where is my performance weakest?
  - what should I do next?
Projects Ideas 2

• Better rule learning algorithm
  – learning from positive data only?
  – accuracy estimates based on resampling?

• Prep phrase attachment
  – How can KB and background statistics be used?

• Co-reference resolution
  – IBM versus Int.Bus.Mach versus it
Projects Ideas 3

• Active learning
  – what questions should I ask in today’s 5 min session with a human?
  – what new data should I download?

• Temporal scoping
  – How can we determine *when in time* a fact holds?
  – Consider earliest web page containing the fact?
  – Date web pages?
  – Read the temporal scope?