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

it's underconstrained!!

Paris
Pittsburgh
Seattle
Cupertino

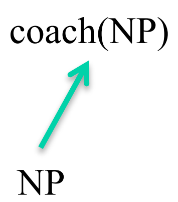
mayor of arg1
live in arg1

San Francisco
Austin
denial

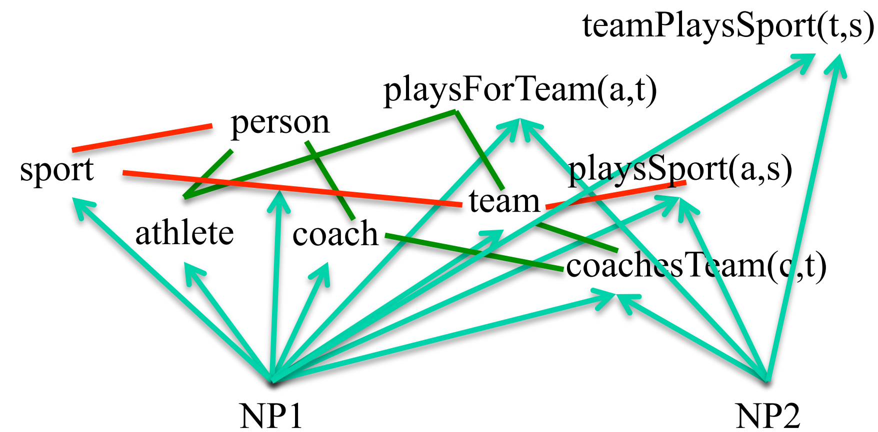
arg1 is home of
traits such as arg1

…

The Key to Accurate Semi-Supervised Learning



Krzyzewski coaches the Blue Devils.



Krzyzewski coaches the Blue Devils.

hard (underconstrained) semi-supervised learning problem

much easier (more constrained) semi-supervised learning problem

Constraining semi-supervised learning 1

Wish to learn $f : X \rightarrow Y$

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

Constraint type 1 (co-training):

if X can be split into redundantly sufficient X1, X2

then learn both $f1: X1 \rightarrow Y$, and $f2: X2 \rightarrow Y$

city? = city?

X1 X2

X: Luke is mayor of Pittsburgh.

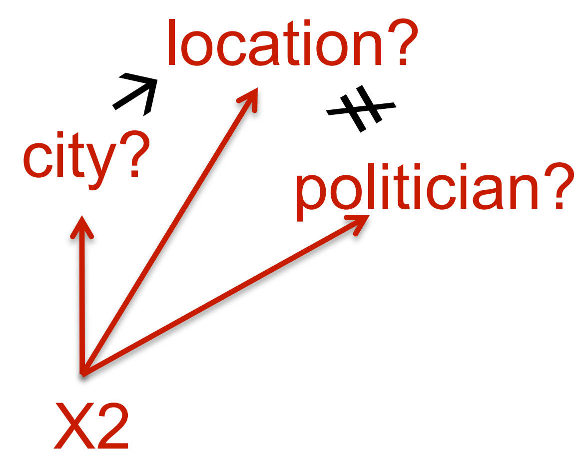
Constraining semi-supervised learning 2

Wish to learn f: X → Y

e.g., city: NounPhraseInSentence → {0,1}

Constraint type 2: couple training of multiple classes

Ontology provides coupling constraints



Luke is mayor of Pittsburgh.

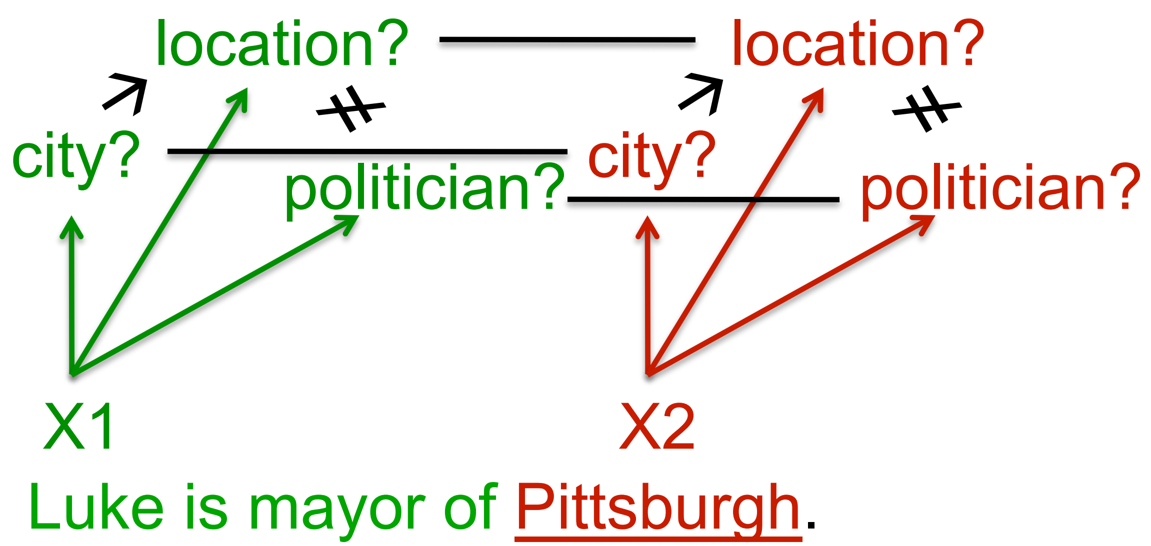
Constraining semi-supervised learning 2

Wish to learn f: X → Y

e.g., city: NounPhraseInSentence → {0,1}

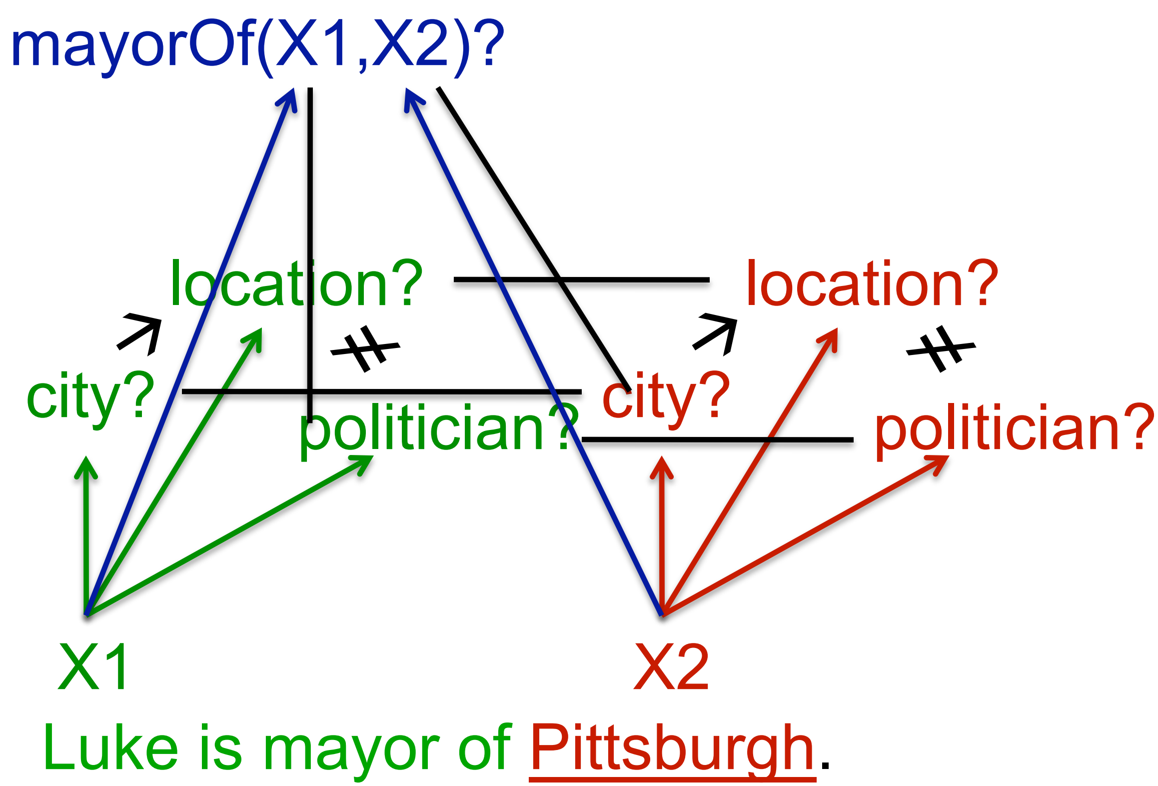
Constraint type 2: couple training of multiple classes

Ontology provides coupling constraints



Constraining semi-supervised learning 3

Constraint type 3 (couple training of <u>classes</u> and <u>relations</u>)



Coupled Bootstrap Learner algorithm

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

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)…

Filtering (M45): CBC || Toronto → Not enough evidence

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

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_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

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

Predicate	5 iterations			10 iterations			15 iterations		
	Full	NS	NCR	Full	NS	NCR	Full	NS	NCR
Actor	93	100	100	93	97	100	100	97	100
Athlete	100	100	100	100	93	100	100	73	100
Board Game	93	76	93	89	27	93	89	30	93
City	100	100	100	100	97	100	100	100	100
Coach	100	63	73	97	53	43	97	47	47
Company	100	100	100	97	90	97	100	90	100
Country	60	40	60	30	43	27	40	23	40
Economic Sector	77	63	73	57	67	67	50	63	40
Hobby	67	63	67	40	40	57	20	23	30
Person	97	97	90	97	93	97	93	97	93
Politician	93	93	97	73	53	90	90	53	87
Product	97	87	90	90	87	100	97	90	77
Product Type	93	93	90	70	73	97	77	80	67
Scientist	100	90	97	97	63	97	93	60	100
Sport	100	90	100	93	67	83	97	27	90
Sports Team	100	97	100	97	70	100	90	50	100
Category Average	92	84	89	82	70	84	83	63	79
Acquired(Company, Company)	77	77	80	67	80	47	70	63	47
CeoOf(Person, Company)	97	87	100	90	87	97	90	80	83
CoachesTeam(Coach, Sports Team)	100	100	100	100	100	97	100	100	90
CompetesIn(Company, Econ. Sector)	97	97	80	100	93	67	97	63	60
CompetesWith(Company, Company)	93	80	60	77	70	37	70	60	43
HasOfficesIn(Company, City)	97	93	40	93	90	27	93	57	30
HasOperationsIn(Company, Country)	100	95	50	100	97	40	90	83	13
HeadquarteredIn(Company, City)	77	90	20	70	77	27	70	60	7
LocatedIn(City, Country)	90	67	57	63	50	43	73	50	30
PlaysFor(Athlete, Sports Team)	100	100	0	100	97	7	100	43	0
PlaysSport(Athlete, Sport)	100	100	27	93	80	10	100	40	30
TeamPlaysSport(Sports Team, Sport)	100	100	77	100	97	80	93	83	67
Produces(Company, Product)	91	83	90	83	93	67	93	80	57
HasType(Product, Product Type)	73	63	17	33	67	33	40	57	27
Relation Average	92	88	57	84	84	48	84	66	42
All	92	86	74	83	76	68	84	64	62

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 .

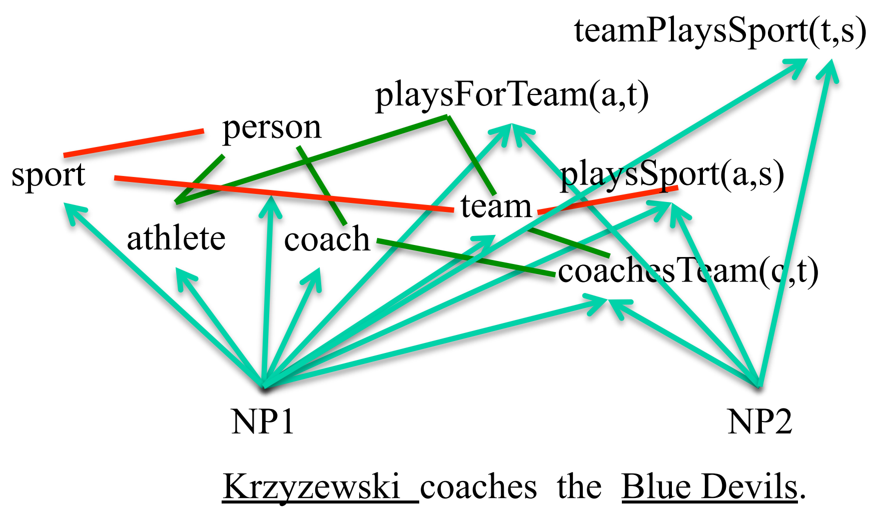
Extending Freebase

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

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?

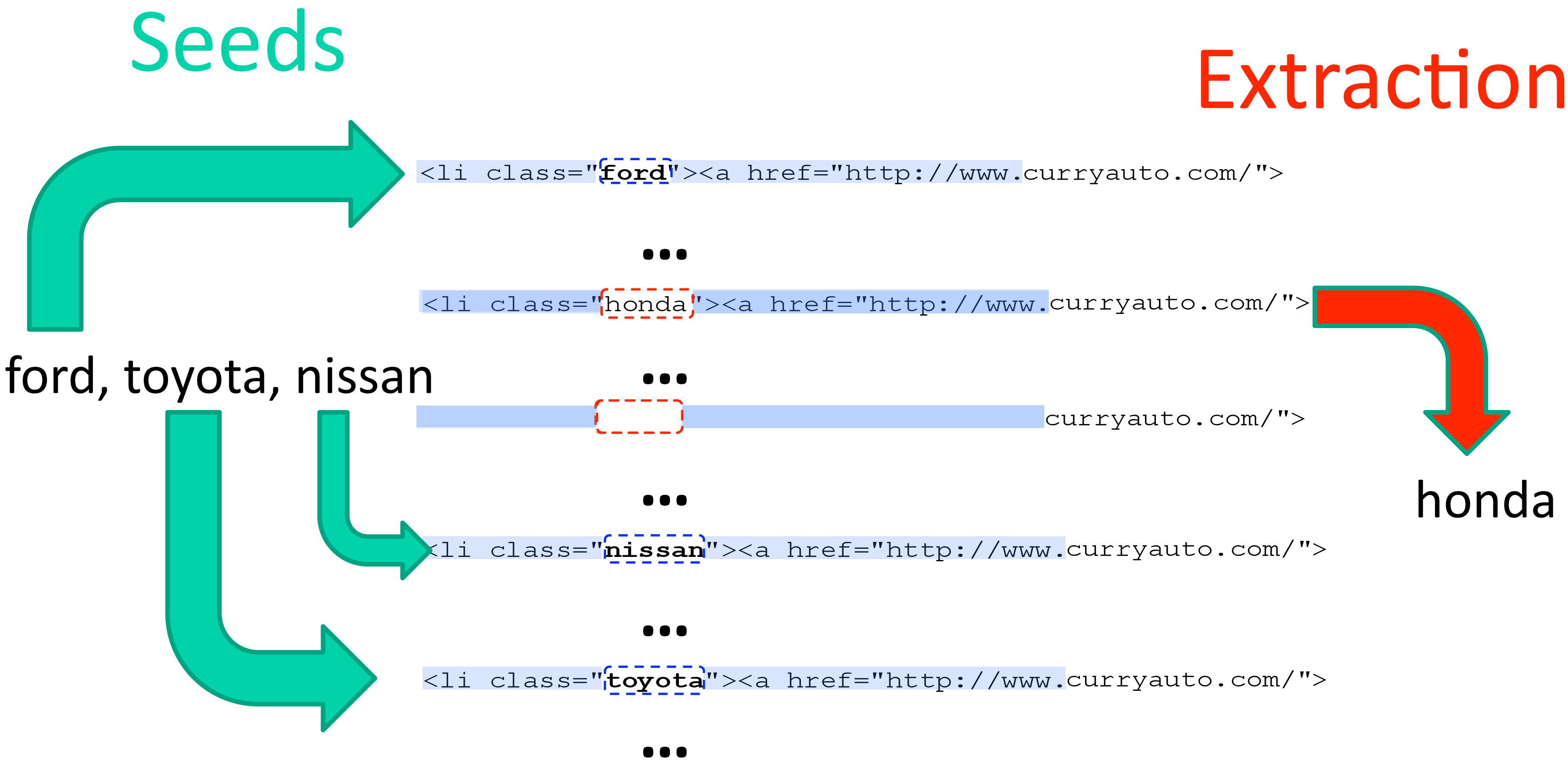


1. incorporate additional learners whose errors will be independent of current learners (e.g., based on HTML)

SEAL

Set Expander for Any Language

*Richard C. Wang and William W. Cohen: Language-Independent Set Expansion of Named Entities using the Web. In *Proceedings of IEEE International Conference on Data Mining* (ICDM 2007), Omaha, NE, USA. 2007.



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:

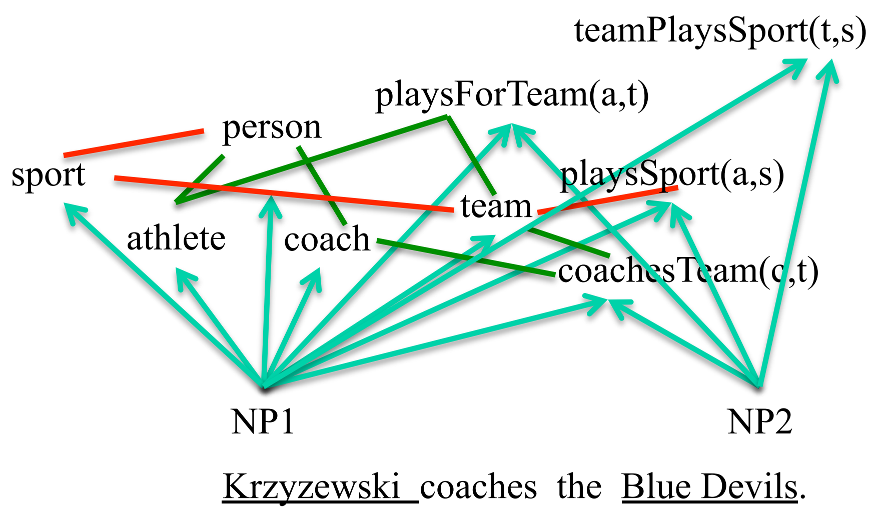
at http://www.transnationale.org/countries/panp.php

pattern: ", <A HREF="../companies/**?X**.php">"

extracts: banco brasil

If the key to accurate self-supervised learning is
<u>coupling the training of many functions</u>,

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)

0.84 playsSport(?x,?y) ← playsFor(?x,?z), teamPlaysSport(?z,?y)

0.70 playsSport(?x,baseball) ← playsFor(?x,cubs)

…

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

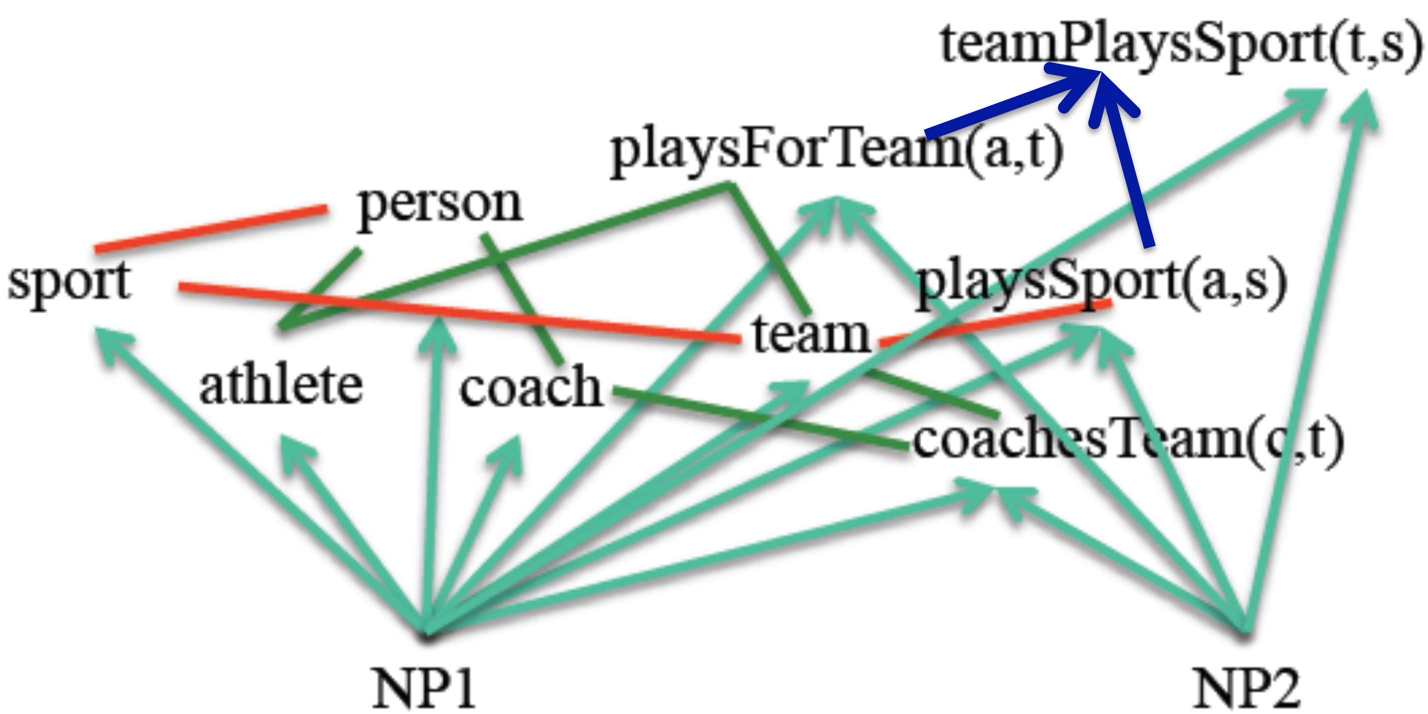
0.70 teamPlaysSport(?x,basketball) ← playsAgainst(?x,pistons)

0.64 teamPlaysSport(?x,?y) ← playsAgainst(?x ?z), teamPlaysSport(?z,?y)

…

Learned Probabilistic Horn Clause Rules

0.81 teamPlaysSport(?x,?y) ← playsForTeam(?x,?z), playSport(?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 <u>decreases</u> 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