How Will We Populate the Semantic Web on a Vast Scale?

Tom M. Mitchell

Weam AbuZaki, Justin Betteridge, Andrew Carlson,
Estevam R. Hruschka Jr., Bryan Kisiel, Burr Settles, Richard Wang

Machine Learning Department
Carnegie Mellon University

October 2009

see: http://rtw.ml.cmu.edu/readtheweb.html

How will we populate the Semantic Web?

1. Humans will enter structured information

2. Database owners will decide to publish theirs

3. Computers will read unstructured web data

this talk

Read the Web: 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

But Natural Language Understanding is Hard!

October 27, 2009 10:04 AM PDT

NASA's Ares I-X test flight delayed by weather

by William Harwood

Font size Print E-mail Share Post a comment

KENNEDY SPACE CENTER, Fla.--Launch of NASA's Ares I-X rocket on a planned $445 million test flight was delayed 24 hours Tuesday because of bad weather and an errant freighter that briefly strayed into the off-shore danger area.

"For everyone, great job today. You gave it a great shot," Launch Director Ed Mango told the team. "We had some opportunities and just couldn't get there, weather didn't cooperate. But good work today."

Launch was rescheduled for 8 a.m. Wednesday. Forecasters are predicting a 60 percent chance of acceptable weather during a four-hour launch window, with lighter winds and less cloud cover. It is not yet clear whether Thursday is an option if additional problems force another delay Wednesday.

NASA began Tuesday's launch campaign at 1 a.m. EDT with the start of a seven-hour countdown. With forecasters concerned about high clouds, showers, and friction-induced static charge

How to make machine reading more plausible

- Leverage *redundancy* on the web
- Target reading to populate a *given ontology*
- Use new *coupled semi-supervised learning* algorithms
- Seed learning using Freebase, DBpedia, …

Read the Web project

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:

- input ontology defining 10^2 classes and relations
- 10-20 seed examples of each

Task:

- learn to extract / extract to learn
- running over 200M web pages, for a week

Result:

- KB with 10^4-10^5 extracted triples

Browse the KB

- ~ 20,000 entities, ~ 40,000 extracted beliefs
- learned from 10-20 seed examples per predicate, 200M unlabeled web pages
- ~ 5 days computation

Initial ontology: Initial ontology

After days of self-supervised learning: populated KB

1. Coupled semi-supervised learning of category and relation extractors

Semi-Supervised Bootstrap Learning

it's underconstrained!!

Extract cities:

Paris
Pittsburgh
Seattle
Cupertino

San Francisco
Austin
denial

anxiety
selfishness
Berlin

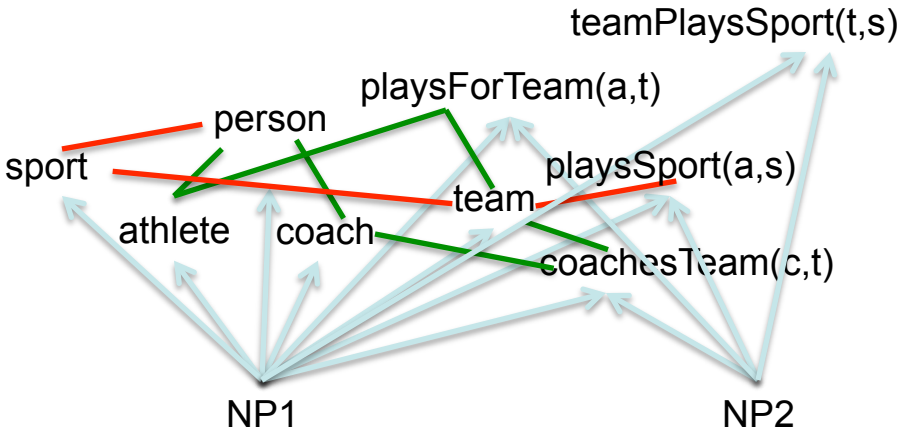
mayor of arg1
live in arg1

arg1 is home of
traits such as arg1

The Key to Accurate Semi-Supervised Learning

Krzyzewski coaches the Blue Devils.

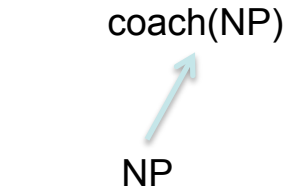
hard (underconstrained) semi-supervised learning problem



Krzyzewski coaches the Blue Devils.

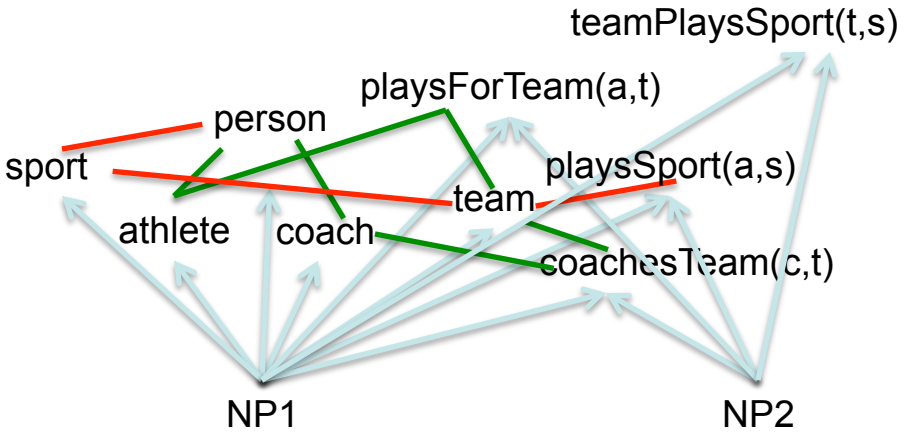
much easier (more constrained) semi-supervised learning problem

The Key to Accurate Semi-Supervised Learning



Krzyzewski coaches the Blue Devils.

hard (underconstrained) semi-supervised learning problem



Krzyzewski coaches the Blue Devils.

much easier (more constrained) semi-supervised learning problem

Key idea: Couple the training of many functions to make unlabeled data more informative.

Coupled training type 1 (co-training)

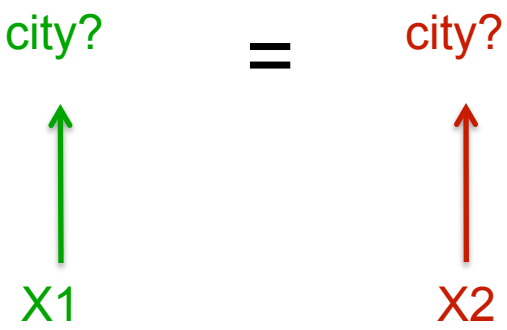
Wish to learn f : X → Y

e.g., city : NounPhrase → {0,1}

Learn 2 functions with different input features

f1: X1 → Y, and f2: X2 → Y

Coupling: force their outputs to agree over unlabeled examples



X: Luke is mayor of Pittsburgh.

Coupled training type 2

Wish to learn f1: X → Y1, f2: X → Y2,

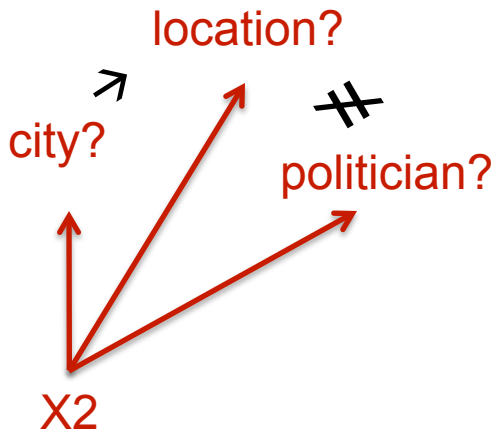
such that: (∀x) g(f1(x), f2(x))

e.g.

location: NounPhrase → {0,1}

politician: NounPhrase → {0,1}

g(y1,y2) = not (and(y1,y2))



Luke is mayor of Pittsburgh.

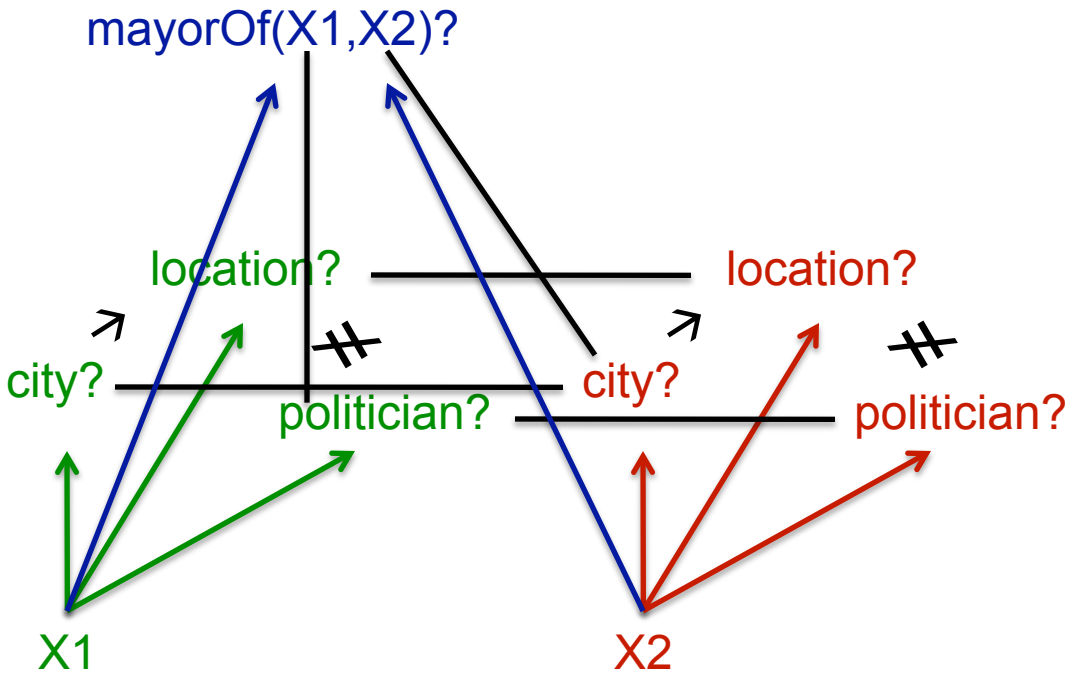
Coupled training type 3

Constraint type 3 (argument type consistency)

mayorOf: NP1 x NP2 → {0,1}

city: NP1 → {0,1}

politician: NP2 → {0,1}



Luke is mayor of Pittsburgh.

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), …
Micron || Boise → arg2 is headquarters for chipmaker arg1, arg1 of arg2, arg1 Corp headquarters in arg2

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

Assessment (M45):
Classify candidate instances with a Naive Bayes classifier
Features related to strength of co-occurrence with each pattern

Promote top ranked instances and patterns. Use type-checking. Score patterns with estimate of precision

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

Automatically extracted companies

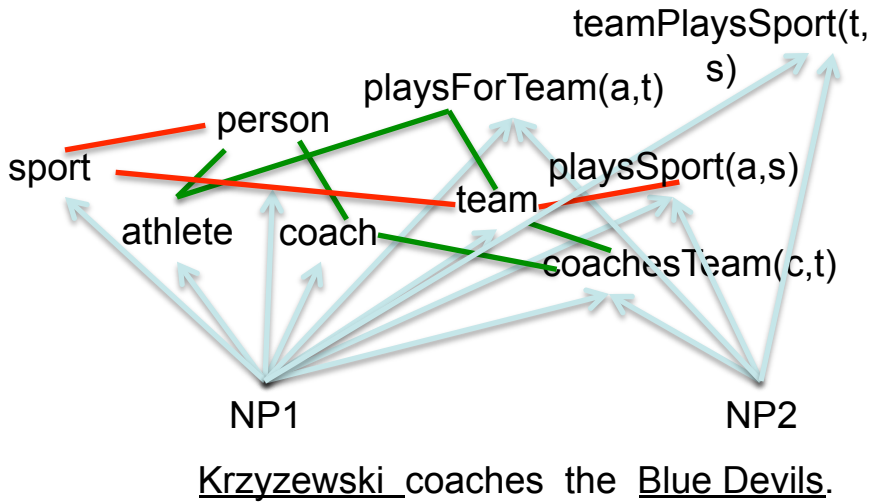
ebay:
  generalizations = {company}
  literalString = {eBay, EBay, Ebay, ebay, EBAY, eBAY}
  acquired = {skype, stumbleupon}
  competesWith = {amazon, yahoo, google, microsoft}
  hasOfficeInCountry = {usa, united_kingdom}

nissan:
  generalizations = {company}
  literalString = {Nissan, NISSAN, nissan}
  acquired = {toyota}
  acquiredBy = renault
  hasOfficeInCountry = {japan, usa, mexico}
  competesWith = {honda}

ibm:
generalizations = {company}
candidateValues = {conference, company, product}
headquarteredIn = armonk
candidateValues = {armonk}
producesProduct = {pc}
candidateValues = {domino, thinkpad_line, ibm_e_business_logo, first_pcs, powerpc, internet, ibm_pc, iseries, rational, first_pc, quickplace, first_ibm_pc, vga_controller, original_pc, at_computer, wsfl_specification, selectric, pc, pc_convertible, workplace_client_technology, workplace, ids, opteron_server, linux_strategy, very_interesting_study, video_graphics_array, business_partner_emblem, ibm, …}
acquired = {iss, cognos, informix}
candidateValues = {spi, watchfire, telelogic, daksh, lotus, iss, internet_security_systems, gluecode, cognos, sequent, tivoli, diligent, informix, webify_solutions, geronimo, rational, information_laboratory, meiosys, webify, …}
acquiredBy = lenovo_group
candidateValues = {lenovo_group, lenovo, china, arsenal}
competesWith = {sun, texas_instruments, samsung, hewlett_packard, apple, novell, oracle, microsoft, ricoh, hp, amazon}
companyEconomicSector = {software}
hasOfficeInCountry = {united_states, canada, usa, germany, england, uk, france}
candidateValues = {san_jose, dallas, cambridge, europe, boca_raton, boulder, united_states, tucson, november, new_york, poughkeepsie, canada, october, united, research_triangle_park, rochester, beaverton, armonk, usa, u_s, germany, new_delhi, boeblingen, england, uk, france, us, facebook, masters_degree}

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

then how can we create even more coupling?

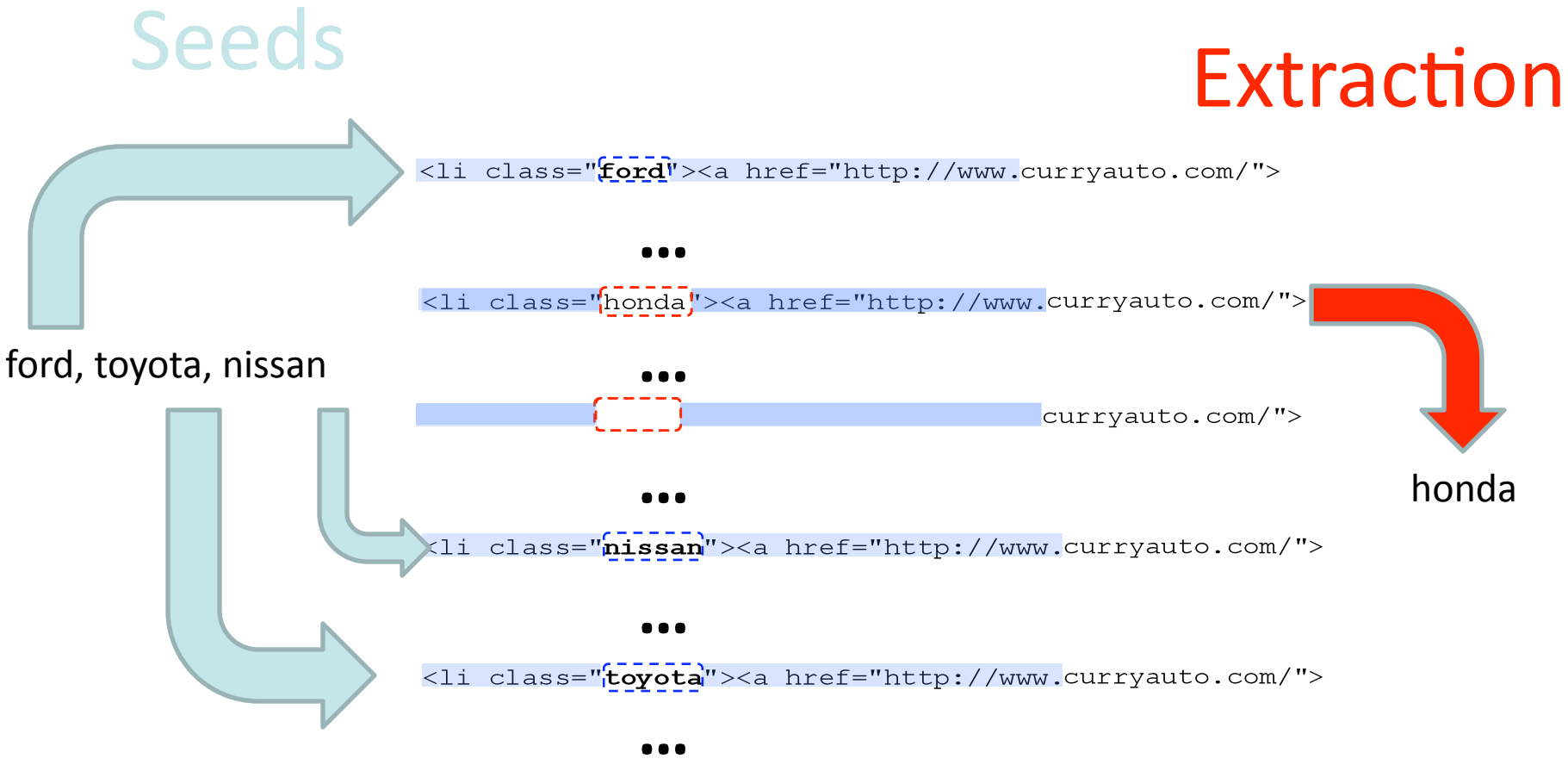


1. introduce additional coupling by adding a learner based on HTML features instead of free text

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

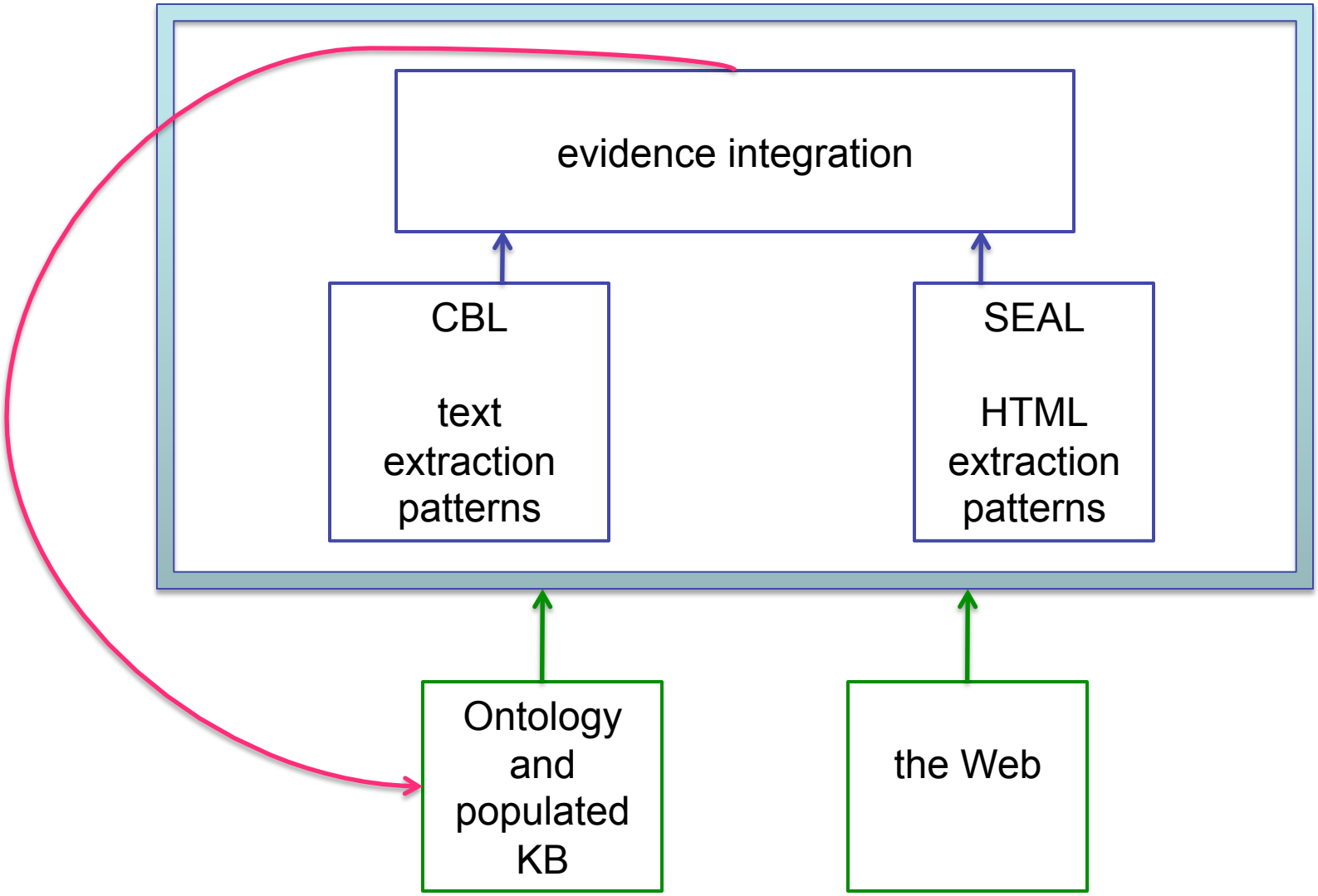
Retrain CBL from current KB, allow it to add to KB

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

Typical learned SEAL extractors:

URL:	http://www.shopcarparts.com/
Wrapper:	.html" CLASS="shopcp">[...] Parts</A>
Content:	acura, audi, bmw, buick, cadillac, chevrolet, chevy, chrysler, daewoo, daihatsu, dodge, ea
URL:	http://www.allautoreviews.com/
Wrapper:	 <a href="auto_reviews/[...]/
Content:	acura, audi, bmw, buick, cadillac, chevrolet, chrysler, dodge, ford, gmc, honda, hyundai, i
URL:	http://www.hertrichs.com/
Wrapper:	<li class="franchise [...]"> <h4><a href="#">
Content:	buick, chevrolet, chrysler, dodge, ford, gmc, isuzu, jeep, lincoln, mazda, mercury, nissan,

Coupled learning of text and HTML patterns



Predicate	Precision (%)					Promoted Instances (#)				
	CPL	UPL	CSEAL	SEAL	MBL	CPL	UPL	CSEAL	SEAL	MBL
AcademicField	70	83	90	97	100	46	903	203	1000	181
Actor	100	33	100	97	100	199	1000	1000	1000	380
Animal	80	50	90	70	97	741	1000	144	974	307
Athlete	87	17	100	87	100	132	930	276	1000	555
AwardTrophyTournament	57	7	53	7	77	86	902	146	1000	79
BoardGame	80	13	70	77	90	10	907	126	1000	31
BodyPart	77	17	97	63	93	176	922	80	1000	61
Building	33	50	30	0	93	597	1000	57	1000	14
Celebrity	100	90	100	100	97	347	1000	72	747	514
CEO	33	30	100	77	100	3	902	322	1000	30
City	97	100	97	87	97	1000	1000	368	1000	603
Clothing	97	20	43	27	97	83	973	167	1000	102
Coach	93	63	100	83	100	188	838	619	1000	242
Company	97	83	100	100	97	1000	1000	245	1000	784
Conference	93	53	97	90	100	95	990	437	928	92
Country	57	33	97	37	93	1000	1000	130	1000	207
EconomicSector	60	23	100	10	77	1000	1000	34	1000	138
Emotion	77	53	87	60	83	483	992	183	1000	211
Food	90	70	97	80	100	811	1000	89	1000	272
Furniture	100	0	57	57	90	55	963	215	1000	95
Hobby	77	33	77	50	90	357	936	77	1000	127
KitchenItem	73	3	88	13	100	11	900	8	960	2
Mammal	83	50	93	50	90	224	1000	154	1000	169
Movie	97	57	97	100	100	718	1000	566	1000	183
NewspaperCompany	90	60	60	97	100	179	1000	1000	1000	241
Politician	80	60	97	37	100	178	990	30	1000	101
Product	90	83	-	77	70	1000	1000	0	999	127
ProductType	73	63	27	63	50	712	1000	31	1000	159
Profession	73	53	-	57	93	916	973	0	1000	171
ProfessionalOrganization	93	63	100	77	87	104	943	58	1000	163
Reptile	95	3	90	27	100	19	912	149	1000	54
Room	64	0	33	7	100	25	913	12	643	3
Scientist	97	30	100	17	100	83	971	928	1000	130
Shape	77	7	7	7	85	43	985	28	733	26
Sport	77	13	63	83	73	283	1000	225	1000	284
SportsEquipment	20	10	57	23	23	58	902	52	1000	174
SportsLeague	100	7	80	27	86	11	901	10	1000	14
SportsTeam	90	30	87	87	87	301	903	864	944	506
Stadium	93	57	53	63	90	102	767	944	1000	343
StateOrProvince	77	63	83	93	77	202	1000	114	1000	161
Tool	40	13	93	90	97	561	1000	713	1000	59
Trait	53	40	52	47	97	234	1000	21	1000	44
University	93	97	100	90	93	1000	1000	961	1000	516
Vehicle	67	30	50	13	77	460	1000	50	1000	98
Average	78	41	78	59	90	360	960	271	976	199
Weighted average	79	42	86	59	91					

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

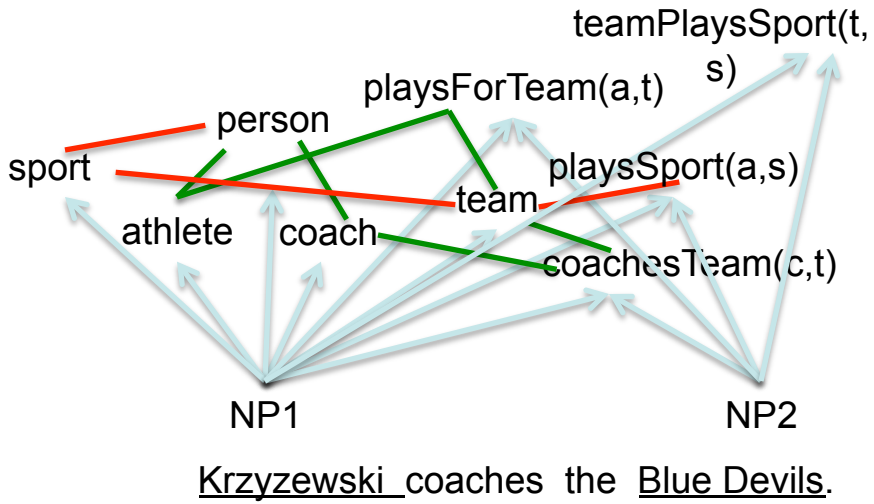
Number of new instances per category

iterations →

Iteration	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
academicField:	100	71	11	7	9	3	0	1	9	10	32	10	17	9	1	1	1	0	5	35	0	0	5	20	20	21	18	9	4	0	0	2	0	0	0	1	0	0	1	0	0	1	0	0
mammal:	18	7	23	21	6	5	4	1	0	0	6	6	8	0	0	0	2	6	1	0	0	0	1	3	0	1	1	2	17	1	1	1	2	0	0	0	0	0	0	0	2	1	0	0
reptile:	0	2	1	0	0	7	3	27	3	3	1	4	4	4	1	0	6	2	3	1	2	18	2	4	3	1	11	7	3	3	1	3	8	2	4	3	4	2	4	5	1	2	4	1
animal:	100	51	4	20	11	4	19	7	0	0	2	1	1	3	14	2	9	1	0	0	4	3	1	0	0	1	0	3	2	30	33	11	10	1	0	0	3	15	5	14	6	6	0	0
awardTrophyT	16	10	8	3	3	3	4	7	7	5	3	14	7	10	0	0	0	1	0	0	0	0	0	0	0	2	4	1	0	1	2	2	4	2	1	0	0	3	13	3	8	1	1	2
boardGame*:	4	3	2	1	0	2	4	4	2	1	0	5	0	0	0	2	0	0	3	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
conference*:	14	12	1	1	2	12	6	4	2	2	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
economicSecto	31	14	37	17	0	2	5	5	4	19	3	0	5	1	0	3	9	0	0	0	0	0	0	0	0	4	3	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
emotion:	80	10	16	12	2	21	3	4	2	4	1	1	2	1	2	7	10	1	1	1	2	22	5	2	6	2	6	1	1	1	1	2	1	1	1	2	5	1	3	3	1	1	2	1
hobby:	29	13	6	3	30	1	1	2	25	5	3	10	0	7	0	0	5	4	8	3	0	2	3	10	5	3	9	0	0	0	0	1	0	4	20	2	1	1	0	0	1	4	0	2
movie:	62	38	19	30	2	3	3	12	0	3	22	0	0	1	0	0	7	4	29	0	6	5	1	1	0	0	0	6	1	7	1	0	0	7	1	0	0	2	3	13	14	24	0	3
productType:	7	21	26	17	21	26	20	16	14	5	23	19	10	5	20	15	12	11	20	22	14	30	11	21	25	12	25	7	14	15	7	11	9	17	13	7	17	8	15	14	28	15	13	8
profession:	2	5	94	25	30	13	12	28	27	1	5	29	30	14	7	4	13	4	2	0	2	5	2	12	13	1	0	0	1	0	0	0	0	0	7	3	7	14	14	6	7	2	1	0
shape:	18	1	0	0	0	0	0	3	0	5	7	7	7	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	3	2	3	0	3	0	0	0	0	0	1
sport:	100	44	28	17	22	8	32	14	7	18	2	4	9	13	1	8	8	6	4	1	1	6	4	2	30	4	5	11	6	0	0	3	1	2	6	1	2	4	2	4	7	3	3	0
trait:	1	3	2	0	0	0	0	0	0	1	0	3	0	0	1	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
company:	100	100	100	100	100	49	33	30	76	46	88	66	36	28	17	22	13	6	21	31	38	16	25	27	46	62	30	31	31	18	16	18	6	37	10	11	46	40	25	16	5	18	2	35
newspaperCom	40	36	75	15	2	13	4	15	16	11	11	1	3	6	1	18	3	11	5	5	14	0	0	6	14	0	2	0	0	1	1	6	6	25	3	3	2	0	0	5	15	3	14	1
professionalOr	16	34	14	17	21	12	6	11	9	5	9	15	5	10	4	6	7	2	4	1	12	8	2	4	3	3	2	4	5	2	6	3	2	9	5	2	8	15	10	6	16	7	2	9
sportsLeague*:	3	1	1	2	0	0	5	6	3	1	0	0	7	0	0	0	0	3	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sportsTeam:	100	100	100	100	65	71	7	64	100	71	81	22	42	39	69	100	100	100	100	100	68	65	21	36	38	15	40	15	36	14	43	8	8	8	12	5	17	50	4	3	17	8	8	44
university:	70	33	22	100	100	57	36	18	8	7	14	38	1	0	0	1	2	12	8	3	9	1	0	1	3	4	3	4	6	6	5	8	0	0	0	22	2	14	12	2	17	4	1	1
athlete:	100	100	66	100	83	29	45	30	100	54	15	51	100	100	45	11	41	100	54	15	53	14	32	21	18	13	25	10	16	9	1	0	1	2	10	22	6	15	22	5	10	18	3	0
celebrity:	99	78	66	67	59	48	35	8	23	17	4	3	14	19	9	11	8	6	13	21	10	16	10	13	10	3	12	6	7	1	3	8	14	10	7	6	13	19	6	8	13	0	20	2
ceo:	50	34	18	8	24	7	19	58	5	23	20	3	25	14	6	17	3	4	3	3	1	1	3	9	15	7	1	1	0	3	0	0	0	1	1	0	2	3	22	5	0	1	5	4
coach:	100	66	25	21	54	28	64	10	5	35	1	4	28	4	2	2	19	3	0	1	1	3	9	6	10	8	1	0	0	0	0	2	4	0	0	0	0	0	0	0	0	0	1	0
politician:	21	13	6	2	4	15	5	7	0	0	0	11	1	0	0	0	12	0	9	7	1	1	1	0	1	0	0	0	3	7	9	4	2	2	0	0	2	4	3	5	0	0	1	0
scientist:	13	11	7	3	14	20	3	14	8	8	6	11	4	9	28	5	6	3	3	5	4	0	0	2	6	6	1	0	1	1	14	5	3	2	0	0	7	5	6	5	8	11	5	3
bodyPart:	15	9	6	3	1	1	1	0	0	1	2	5	0	0	0	1	0	0	0	0	1	2	1	1	1	0	1	0	0	2	0	1	0	0	0	1	0	0	0	1	0	0	1	0
stadium:	95	31	32	33	18	45	5	11	15	17	26	1	3	6	3	10	5	11	8	15	18	0	2	10	94	1	1	11	5	3	32	1	4	18	7	2	5	13	1	3	2	0	1	2
building:	7	12	2	3	2	2	4	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
clothing:	22	14	6	1	3	1	0	1	4	15	2	1	1	0	1	2	0	2	6	1	0	0	2	1	0	0	2	7	1	1	0	0	3	3	10	8	0	0	0	0	0	0	0	0
food:	49	79	47	28	18	9	6	27	2	14	2	6	7	8	6	0	2	4	4	6	8	1	0	1	2	5	5	9	3	3	1	4	19	8	5	7	3	0	1	0	0	0	0	0
furniture:	37	4	3	3	6	7	8	14	3	3	6	4	0	2	1	2	0	1	4	2	30	3	3	0	0	0	0	1	0	1	0	0	0	3	5	0	0	1	1	0	1	0	3	0
kitchenItem:	0	1	0	2	0	0	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
product:	11	8	7	0	2	12	6	25	2	10	12	1	0	0	0	3	11	1	0	1	2	10	2	3	3	17	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
sportsEquipme	20	9	2	8	15	19	26	18	43	16	29	37	31	5	38	14	15	13	27	13	21	14	21	38	15	27	21	15	30	20	20	25	37	19	22	17	26	37	14	28	39	38	16	29
tool:	15	9	3	2	3	5	9	3	2	5	1	1	0	2	10	4	1	1	0	1	1	1	2	0	1	1	0	0	0	1	0	0	0	0	0	5	1	0	0	1	0	0	0	0
vehicle:	10	35	9	2	5	2	1	2	10	11	3	5	0	0	0	0	0	0	2	10	12	1	1	0	0	0	1	0	2	3	1	3	2	1	2	1	1	3	8	4	1	1	1	1
city:	100	100	100	100	100	100	100	100	100	46	48	54	31	44	33	28	29	20	42	58	32	51	29	77	20	24	30	6	7	24	6	6	43	39	7	27	4	41	17	6	48	5	20	13
country:	93	25	4	1	6	15	5	18	1	4	1	1	0	0	2	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	2	0	1	1	0	1	2	0	0	0	0	1
room:	1	2	3	2	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	1	0	0	1	0	0	0	0	0	0	2
stateOrProvinc	22	24	8	21	7	6	3	0	0	4	9	3	6	6	1	1	5	2	5	0	2	2	0	3	1	1	2	20	7	0	0	1	1	1	3	3	8	1	2	18	2	2	1	6

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

then how can we create even more coupling?



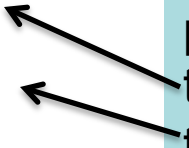
2. allow learner to discover new coupling constraints (by mining its extracted beliefs)

Learning rules by mining the extracted KB

For each relation (e.g., teamPlaysSport(<team>)=<sport>), seek rules to infer its values

- Positive examples: extracted beliefs in the KB examples
- Negative examples: ???

- Ontology to the rescue:
  numberOfValues(teamPlaysSport) = 1
  numberOfValues(competesWith) = any



can infer negative examples from positive for this, but not for this

Some learned rules (out of 49)

{athletePlaysInLeague ?x ?y} ← {athletePlaysForTeam ?x ?z} {teamPlaysInLeague ?z ?y} 0.83 25 2 132

{athletePlaysSport ?x basketball} ← {athletePlaysInLeague ?x nba} 0.96 59 0 18

{cityLocatedInState ?x ?y} ← {cityCapitalOfState ?x ?y} 0.86 40 4 31

{stadiumLocatedInCity ?x ?y} ← {stadiumHomeTeam ?x ?z} {teamPlaysInCity ?z ?y} 0.60 27 16 8

{stateLocatedInCountry ?x ?y} ← {stateHasCapital ?x ?z} {cityLocatedInCountry ?z ?y} 0.77 17 3 7

{teamPlaysInLeague ?x nfl} ← {teamWonTrophy ?x super_bowl} {teamPlaysSport ?x football} 0.92 23 0 2

{teamPlaysSport ?x hockey} ← {generalizations ?x sportsteam} {teamWonTrophy ?x stanley_cup} 0.89 1

{teamPlaysSport ?x baseball} ← {teamPlaysAgainstTeam ?x yankees} 0.87824 13 0 3

{teamPlaysSport ?x ?y} ← {teamPlaysAgainstTeam ?x ?z} {teamPlaysSport ?z ?y} 0.8717 138 18 54

{teamPlaysSport ?x basketball} ← {teamWonTrophy ?x nba_championship} 0.86088 11 0 1

Some embarrassing learned rules

{teamPlaysInLeague ?x nba} ← {teamPlaysSport ?x basketball} 0.94 35 0 35

{cityCapitalOfState ?x ?y} ← {cityLocatedInState ?x ?y} {teamPlaysInLeague ?y nba} 0.80 16 2 23

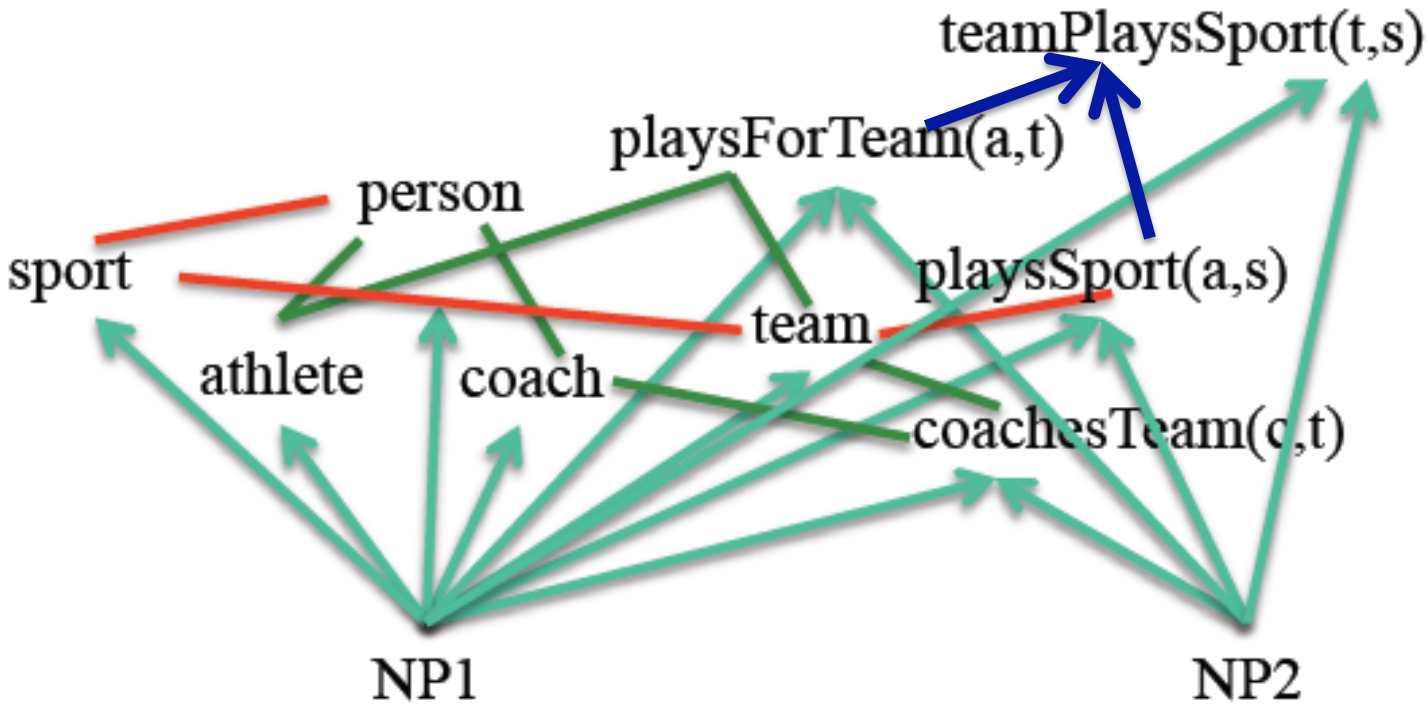
{stateLocatedInCountry ?x united_states} ← {generalizations ?x stateOrProvince} 0.62 21 11 1

Overall impact:

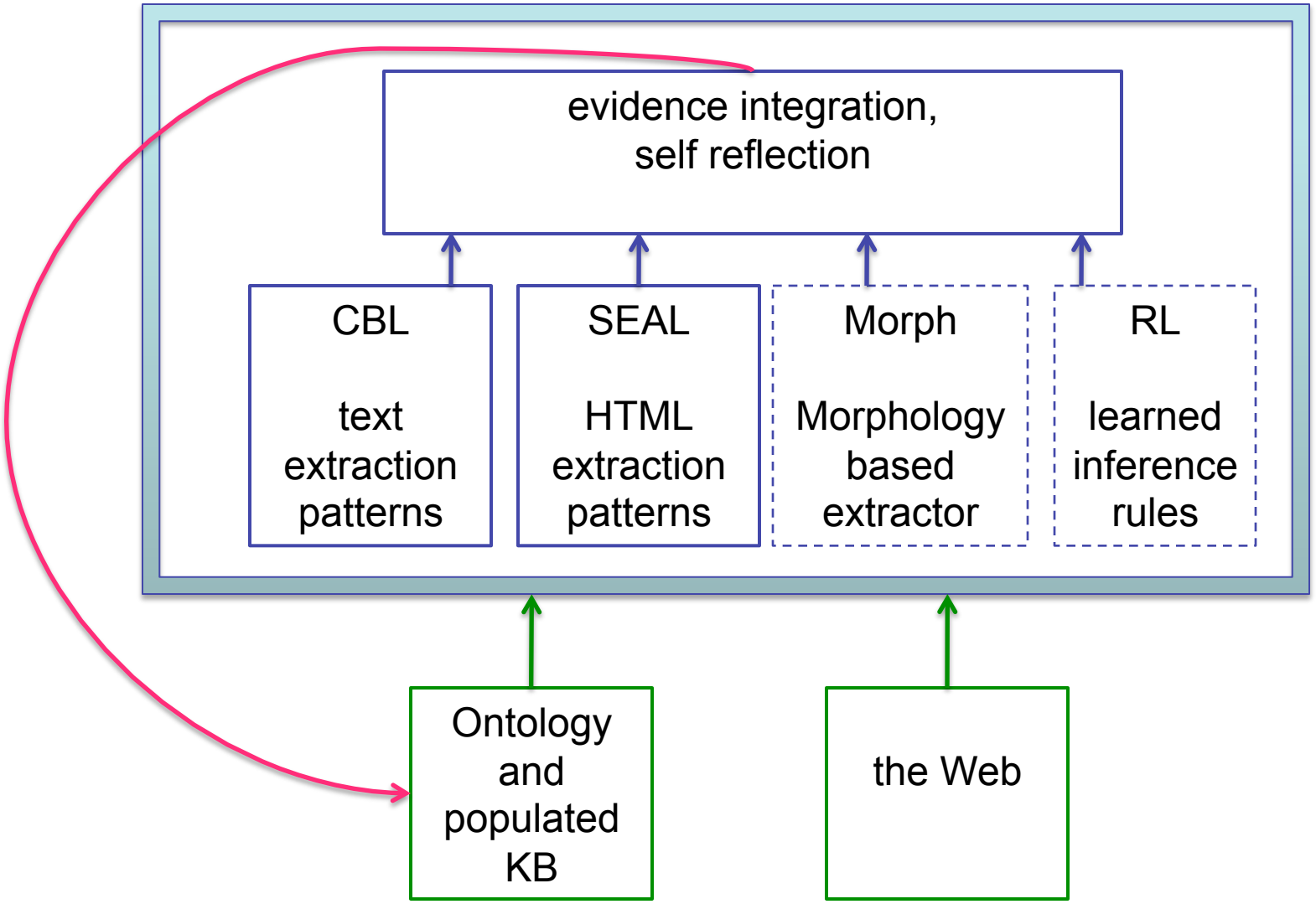
- 49 learned rules
- 15 rules filtered out manually
- remaining rules inferred >1000 new triples that had not been read

Learned Probabilistic Horn Clause Rules

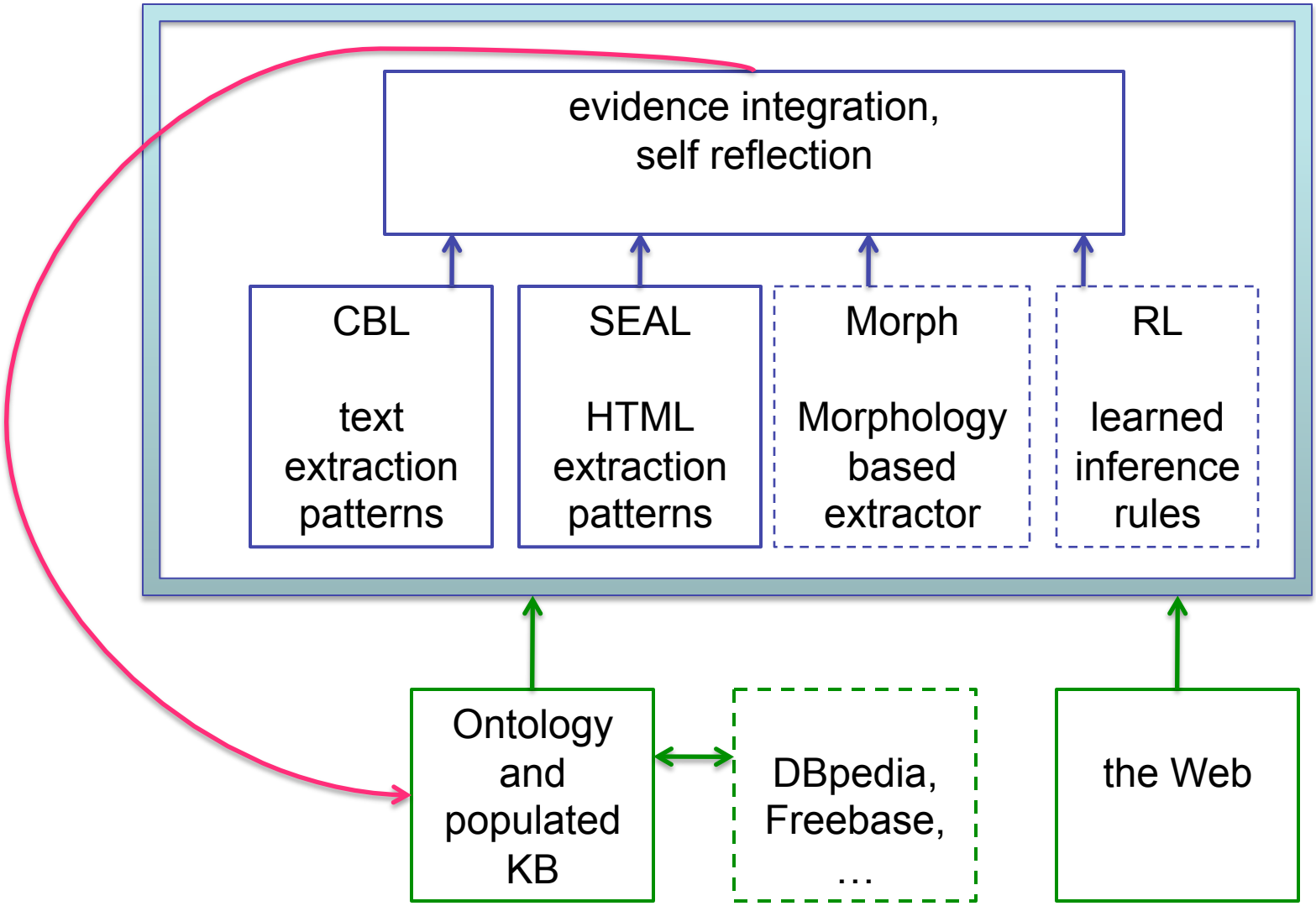
0.81 teamPlaysSport(?x,?y) ← playsForTeam(?x,?z), playSport(?z,?y)



Future steps



Future steps



Summary

- Macro-reading the web can help populate semantic web
  - especially for frequently-mentioned knowledge

- Key design choices:
  - Macro, not micro-reading
  - Coupling the learning of many, many extractors
  - Use target ontology to focus reading, constrain learning

- Next:
  - couple to DBpedia, Freebase, …
  - token/entity distinction
  - self-reflection and never-ending learning

thank you!