Toward an Architecture for Never-Ending Language Learning

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell

School of Computer Science
Carnegie Mellon University

Humans learn many things, for many years,
and become better learners over time.

Why not machines?

Never-Ending Learning

Task: acquire a growing competence without asymptote

- over years
- learning multiple functions
- where learning one thing improves ability to learn the next
- acquiring data from humans, environment

Many candidate domains

- Robots
- Softbots
- Game players

NELL: Never-Ending Language Learner

Inputs:

- Initial ontology
- Handful of examples of each predicate in the ontology
- The web
- Occasional interaction with human trainers

Task:

- Run 24x7, forever
- Each day:
  - Extract more facts from the web to populate initial ontology
  - Learn to read better than yesterday

Ontology

123 Categories

City

Country

Athlete

Company

Sports Team

Economic Sector

Emotion

55 Relations

LocatedIn

HeadquarteredIn

PlaysFor

TeamInLeague

PlaysSport

OperatesInEconomicSector

Why do this?

- Case study in never-ending learning
- Potential for new breakthroughs in natural language understanding
- Producing the world's largest structured KB

Bootstrapped Pattern Learning (Brin 98, Riloff and Jones 99)

Canada
Egypt
France

Pakistan
Sri Lanka
Argentina

Planet Earth
North Africa
Student Council

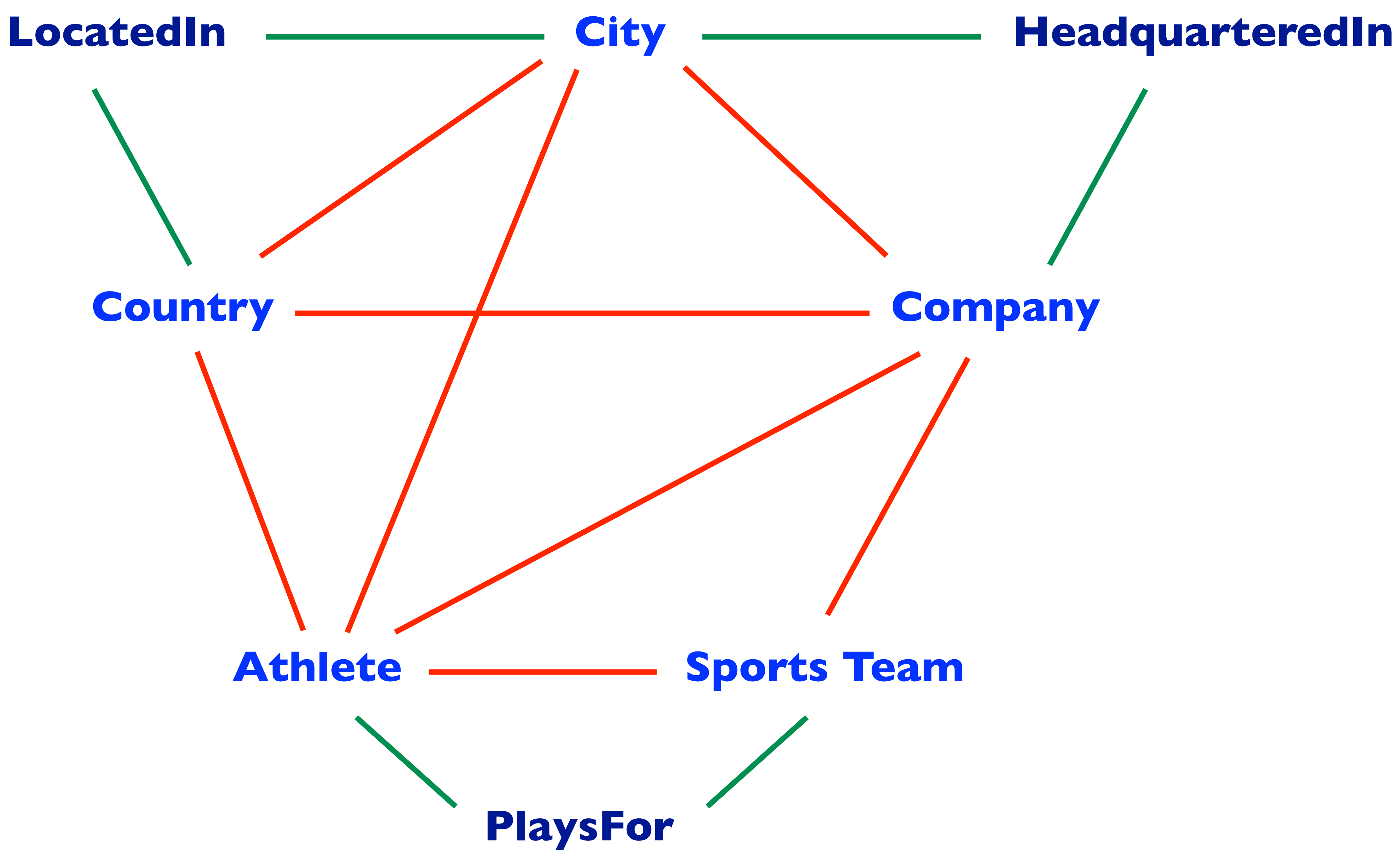
X is the only country
home country of X

invasion of X
elected president of X

Without proper constraints, a never-ending bootstrap learner will "run off the rails."

How can we avoid this?

Solution Part 1: Coupled Learning of Many Functions



Exploiting Mutual Exclusion

Positives:
Canada
Egypt
France

invasion of X
elected president of X

Planet Earth
North Africa
Student Council

Negatives:
Europe
London
Florida
Baghdad
...

nations like X
countries other than X

Pakistan
Sri Lanka
Argentina

Coupled Pattern Learner: Type Checking

Pillar, San Jose

OK

Type Checking Arguments:

... companies such as Pillar ...

... cities like San Jose ...

X , which is based in Y

inclined pillar, foundation plate

Not OK

Solution Part 2: Multiple Extraction Methods

Textual Extraction Patterns

- Mayor of X

List Extraction

- http://www.citymayors.com/statistics/largest-cities-mayors-1.html

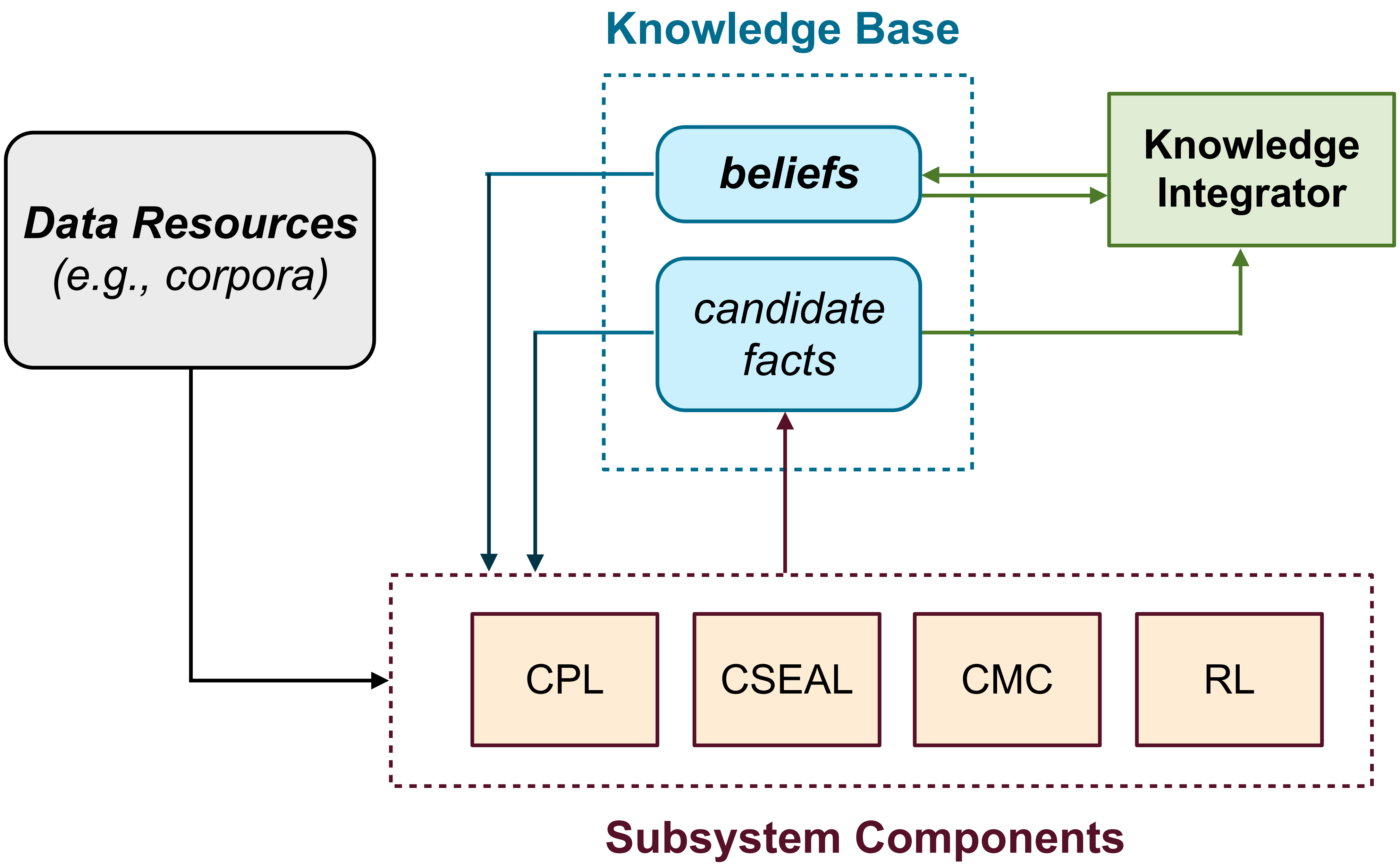
Morphology Classifier

- "-son" suffix likely to be a last name

Rule Learner

- An athlete who plays for a team that plays in the NBA plays in the NBA

NELL architecture



Learned Extraction Patterns

Pattern	Predicate
blockbuster trade for X	athlete
airlines , including X	company
personal feelings of X	emotion
X announced plans to buy Y	companyAcquiredCompany
X learned to play Y	athletePlaysSport
X dominance in Y	teamPlaysInLeague

Example Morphological Features

Predicate	Feature	Weight
mountain	LAST=peak	1.791
mountain	LAST=mountain	1.093
mountain	FIRST=mountain	-0.875
musicArtist	LAST=band	1.853
musicArtist	POS=DT_NNS	1.412
musicArtist	POS=DT_JJ_NN	-0.807

Example Learned Rules

- Athletes who play in the NBA play basketball.
- Teams that won the Stanley Cup play in the NHL.
- If an athlete plays for a team that plays in a league, then the athlete plays in that league.

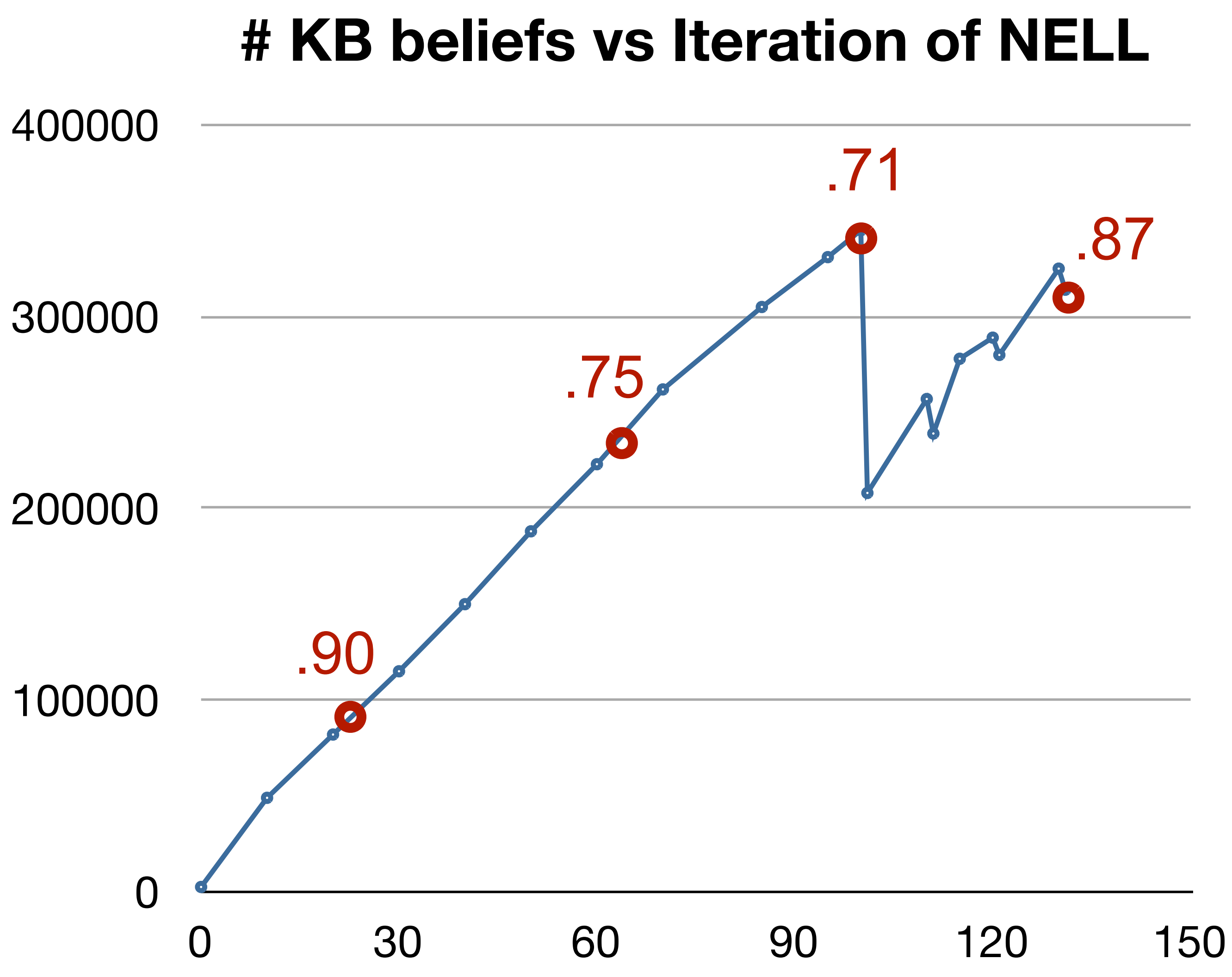
(Solution Part 3: Discovery of New Constraints)

6 facts learned in the last week

Predicate	Instance
architect	Charles Moore ✓
park	Parque Nacional Conguillio ✓
kitchen item	oven safe skillet ✓
county	Woodbury County ✓
card game	cash bonus ✗
perception event	energy engineering ✗

NELL right now

- 314K beliefs
- 30K textual extraction patterns
- 486 accepted learned rules leading to 4K new beliefs
- 65-75% of predicates currently populating well, others are receiving significant correction



Lessons so far

- Key architectural ingredients:
  - Coupled target functions
  - Multiple extraction methods
  - Discovery of new constraints among relations
- We've changed the accuracy vs. experience curve from  to  , but not to 

The future

- Distinguish entities from textual strings
- More human involvement
- Ontology extension
- Planning

Thank you

Thanks to Yahoo! for M45 computing

Thanks to Jamie Callan for ClueWeb09 corpus

Thanks to Google, NSF, and DARPA for partial funding

Learn more at http://rtw.ml.cmu.edu