Never-Ending Learning

ICML 2019 Tutorial

Tom Mitchell
Carnegie Mellon University
tom.mitchell@cs.cmu.edu

Partha Talukdar
IISc Bangalore and KENOME
ppt@iisc.ac.in

https://sites.google.com/site/neltutorialicml19/

Long Beach, June 10, 2019

Motivation:

We will never really understand learning until we build machines that, like people:

- learn many different things,
- from years of diverse experience,
- in a staged, curricular fashion,
- and become better learners over time.

Much research over the years…

- Learning to learn
- Life-long learning
- Never Ending Learning

Essentially the same goal:

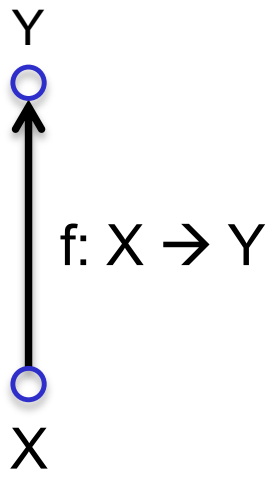
- learn many different things,
- from years of diverse experience,
- in a staged, curricular fashion,
- and become better learners over time.

Many related subproblems…

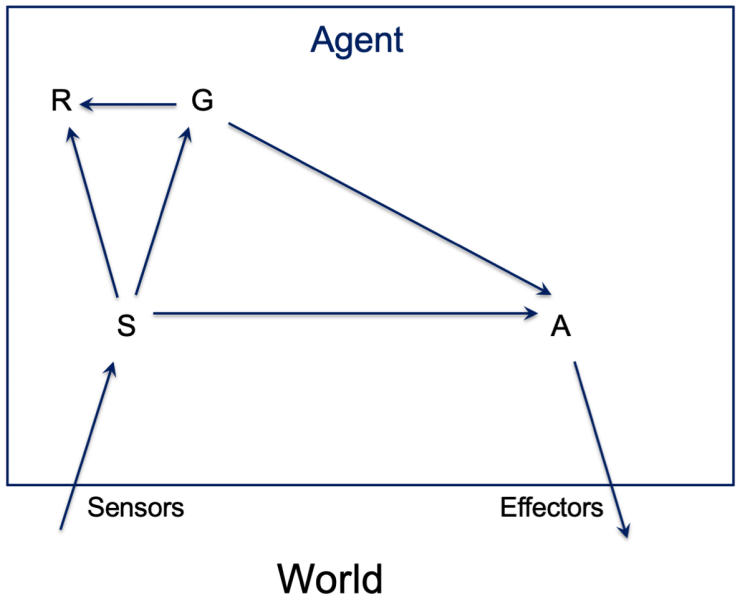
- Multi-task learning
- Curriculum learning
- Cross-task knowledge transfer
- Meta-learning
- Amortized representation learning
- Curiosity-driven learning
- Multi-agent learning
- Cognitive modeling
- …

Fundamentally a question of agent architecture

Learning single function:



Learning agent:



Fundamentally a question of agent architecture

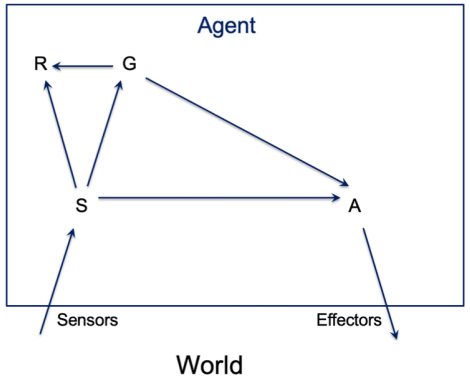
What set of functions, memories, drives/rewards should architecture have?

How should they be interconnected?

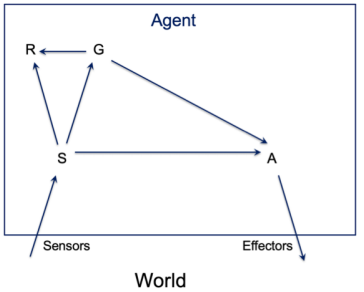
What self-reflection and learning mechanisms?

What knowledge should be represented by explicit functions/mappings/memories, vs. implicit, computed on demand?

…



What should a theory of Learning Agents answer?



might model learning agent A as tuple <S,E,M,F,G,L>

- S = sensors
- E = effectors
- F = set of functions
- M = set of memory units
- G = graph specifying data flow among F, M, S, E
- L = learning mechanism

might model L as another agent L = $<S_L,E_L,M_L,F_L,G_L>$

- where S_L, E_L sense and act on Agent, especially its F, M, G

What should a theory of Never Ending Learning Agents answer?

A = <Sensors, Effectors, Memory, Fns, Graph, L>

L = <S_L,E_L,M_L,F_L,G_L>

Q: What initial A structure <S,E,M,F,G,L> suffices to ensure agent A can in principle modify itself into any computable behavior with respect to its sensors S and effectors E?

Q: What initial A structure allows A to learn from unlabeled data?

Q: What initial A structure allows A to learn to learn?

Q: What initial A structure allows A to self-reflect on its own abilities, and redirect its learning effort?

A Case Study: NELL

NELL: Never-Ending Language Learner

The Learning Agent task:

- run 24x7, forever
- each day:
    1. extract more facts from the web to populate knowledge base
    2. learn to read (perform #1) better than yesterday

Inputs:

- initial ontology (categories and relations)
- dozen examples of each ontology predicate
- the web
- occasional interaction with human trainers

NELL's Eight Years

Ran 24x7, from January, 2010 to September 2018.

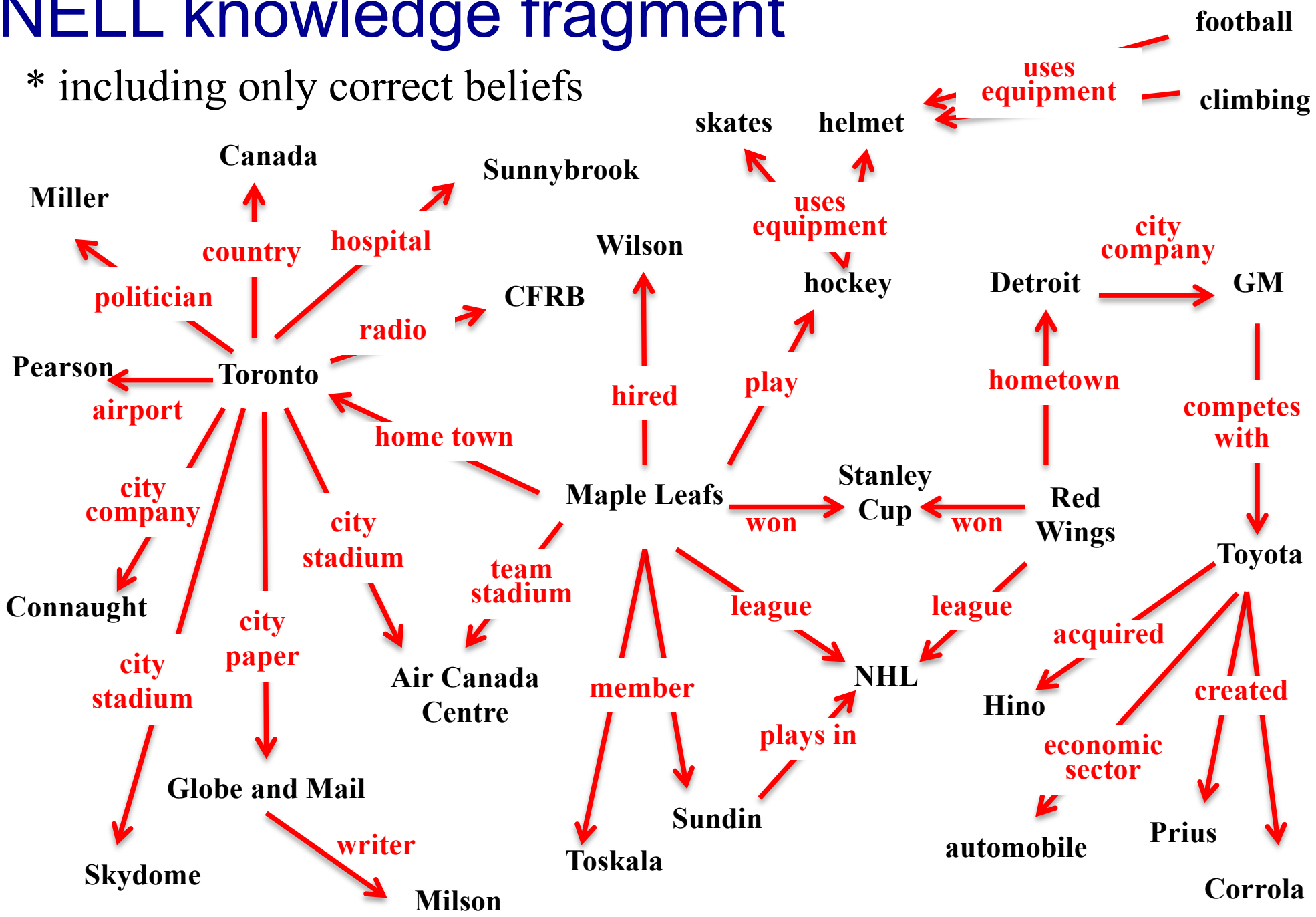
Result:

- KB with ~120 million confidence-weighted beliefs
- learned to improve its reading ability
  its reasoning ability
  its learning ability
- extended its ontology of known relations

Case study of never-ending learning agent

NELL knowledge fragment

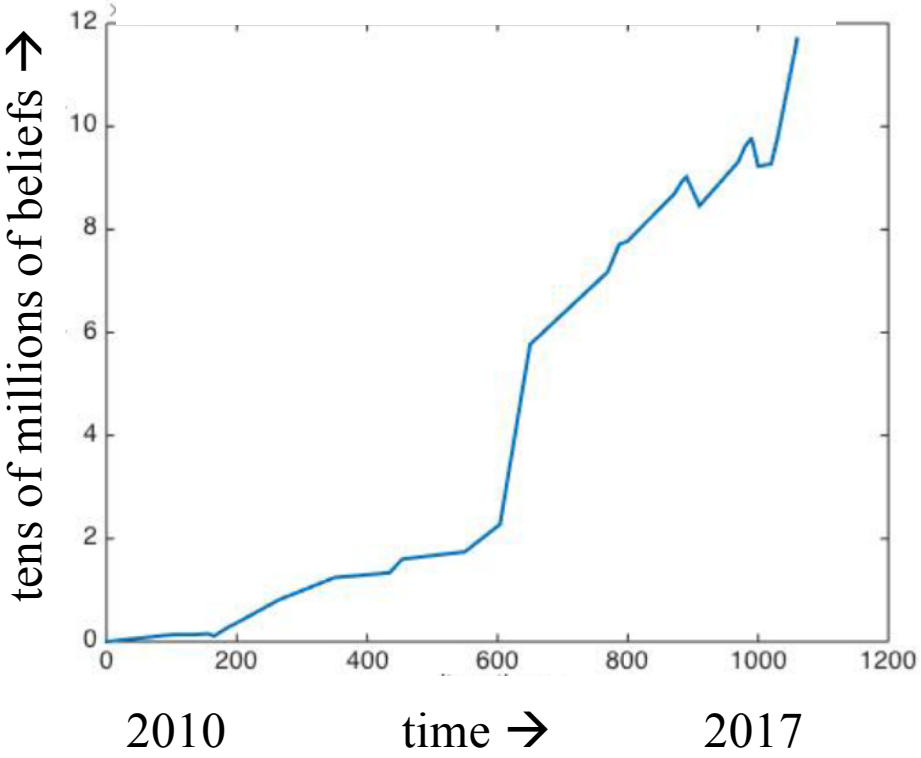
* including only correct beliefs



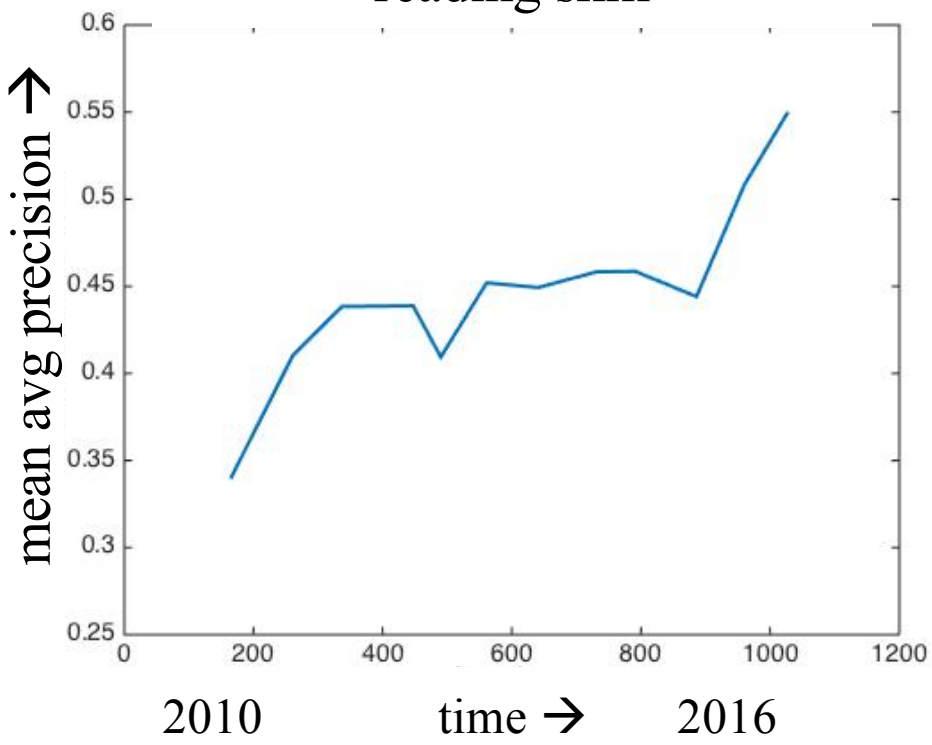
NELL Improving Over Time

[Mitchell et al., CACM 2018]

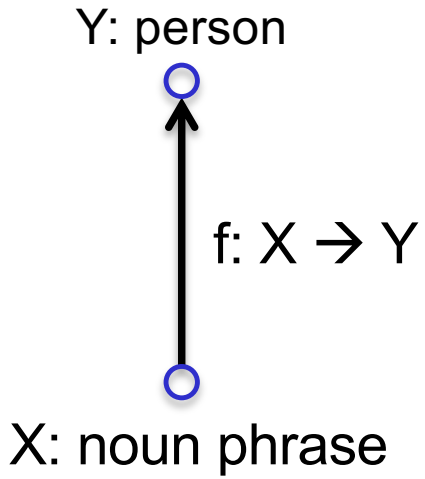
10's of millions of beliefs



reading skill

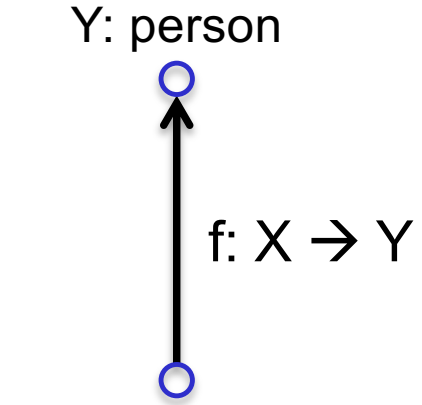


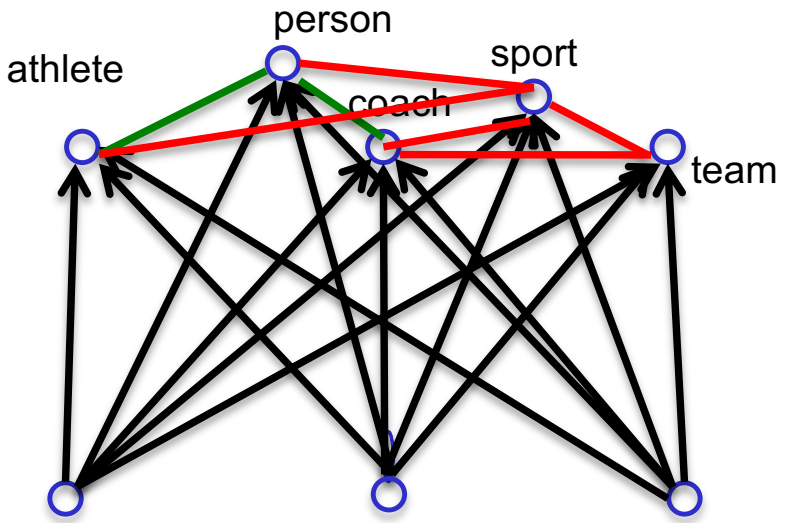
Q: What initial A structure allows A to learn from unlabeled data?



hard
(underconstrained)
semi-supervised
learning

Key Idea: Massively coupled semi-supervised training





noun phrase
text context
"__ is my son"

noun phrase
morphology
ends in '…ski'

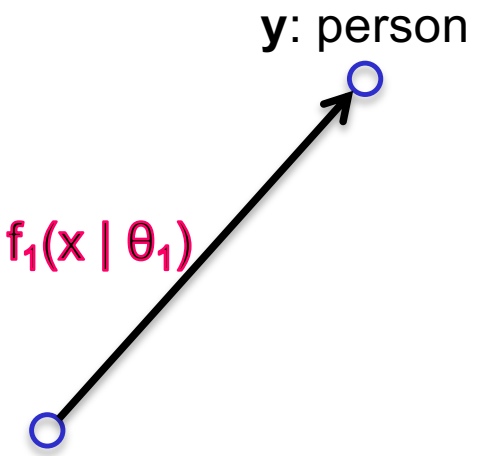
noun phrase
URL specific
appears in list2
at URL35401

hard
(underconstrained)
semi-supervised
learning

much easier
(more constrained)
semi-supervised
learning

Supervised training of 1 function:

$$\theta_1 = \arg\min_{\theta_1} \sum_{\langle x,y \rangle \in \textit{labeled data}} |f_1(x|\theta_1) - y|$$



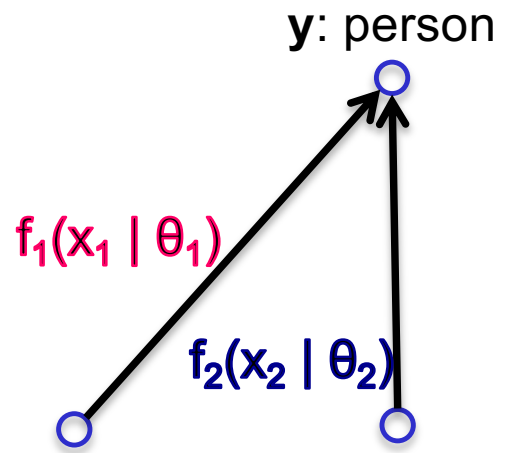
x:

NP context distribution

__ is a friend
rang the __
...
__ walked in

Coupled training of 2 functions:

$$\theta_1, \theta_2 = \arg\min_{\theta_1,\theta_2} \sum_{\langle x,y\rangle \in \text{labeled data}} |f_1(x|\theta_1) - y| + \sum_{\langle x,y\rangle \in \text{labeled data}} |f_2(x|\theta_2) - y| + \sum_{x \in \text{unlabeled data}} |f_1(x|\theta_1) - f_2(x|\theta_2)|$$



__ is a friend
rang the __
...
__ walked in

capitalized?
ends with '...ski'?
...
contains "univ."?

NELL Learned Contexts for “Hotel” (~1% of total)

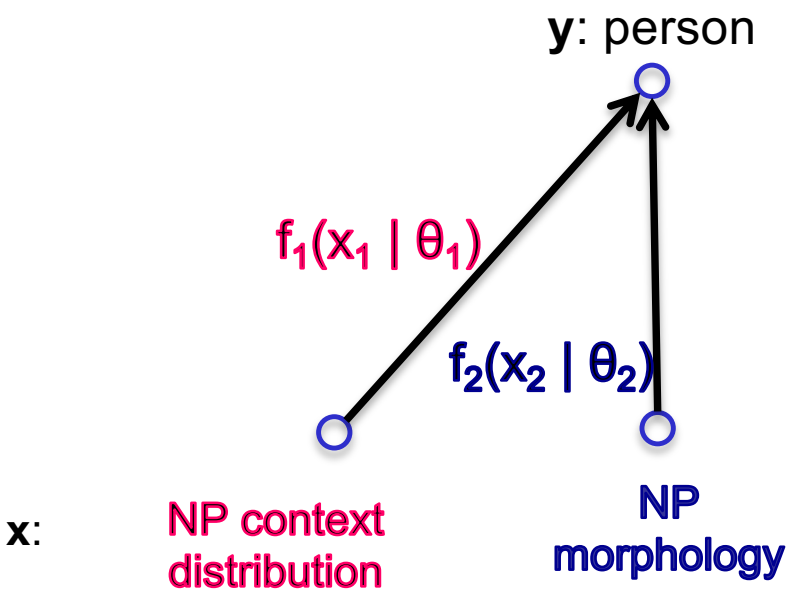
"_ is the only five-star hotel” "_ is the only hotel” "_ is the perfect accommodation" "_ is the perfect address” "_ is the perfect lodging” "_ is the sister hotel” "_ is the ultimate hotel" "_ is the value choice” "_ is uniquely situated in” "_ is Walking Distance” "_ is wonderfully situated in” "_ las vegas hotel” "_ los angeles hotels” "_ Make an online hotel reservation” "_ makes a great home-base” "_ mentions Downtown” "_ mette a disposizione” "_ miami south beach” "_ minded traveler” "_ mucha prague Map Hotel” "_ n'est qu'quelques minutes” "_ naturally has a pool” "_ is the perfect central location” "_ is the perfect extended stay hotel” "_ is the perfect headquarters” "_ is the perfect home base” "_ is the perfect lodging choice" "_ north reddington beach” "_ now offer guests” "_ now offers guests” "_ occupies a privileged location” "_ occupies an ideal location” "_ offer a king bed” "_ offer a large bedroom” "_ offer a master bedroom” "_ offer a refrigerator” "_ offer a separate living area" "_ offer a separate living room” "_ offer comfortable rooms” "_ offer complimentary shuttle service” "_ offer deluxe accommodations” "_ offer family rooms” "_ offer secure online reservations” "_ offer upscale amenities” "_ offering a complimentary continental breakfast” "_ offering

NELL Highest Weighted* string fragments: “Hotel”

1.82307	SUFFIX=tel
1.81727	SUFFIX=otel
1.43756	LAST_WORD=inn
1.12796	PREFIX=in
1.12714	PREFIX=hote
1.08925	PREFIX=hot
1.06683	SUFFIX=odge
1.04524	SUFFIX=uites
1.04476	FIRST_WORD=hilton
1.04229	PREFIX=resor
1.02291	SUFFIX=ort
1.00765	FIRST_WORD=the
0.97019	SUFFIX=ites
0.95585	FIRST_WORD=le
0.95574	PREFIX=marr
0.95354	PREFIX=marri
0.93224	PREFIX=hyat
0.92353	SUFFIX=yatt
0.88297	SUFFIX=riott
0.88023	PREFIX=west

* By logistic regression

Type 1 Coupling: Co-Training, Multi-View Learning



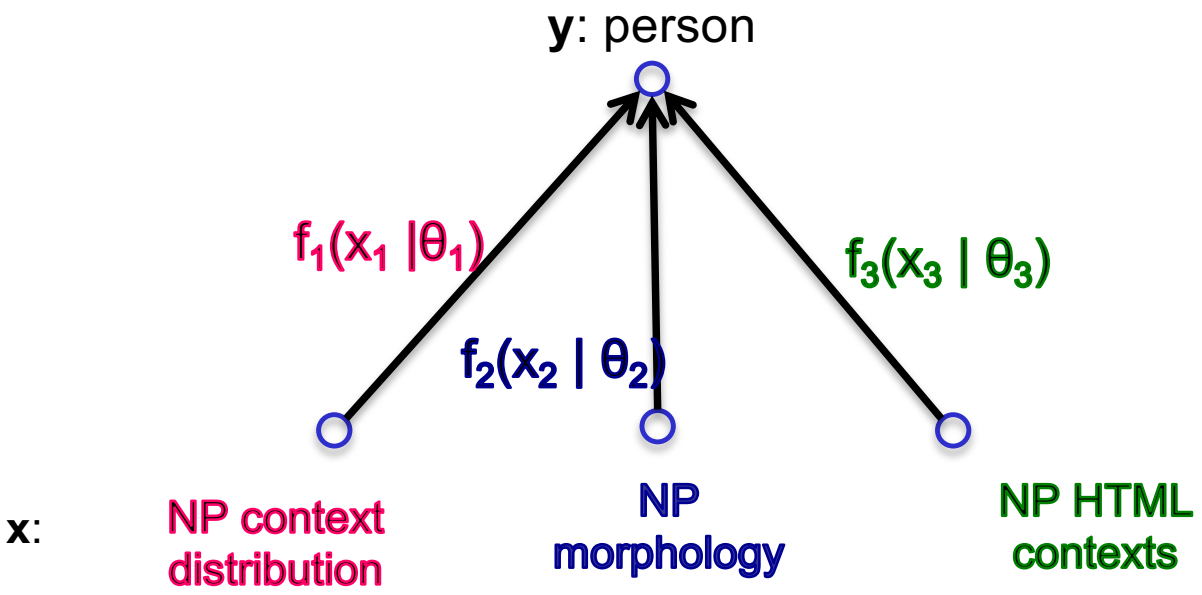
Theorem (Blum & Mitchell, 1998):

If f_1,and f_2 are PAC learnable from noisy *labeled* data, and X_1, X_2 are conditionally independent given Y,

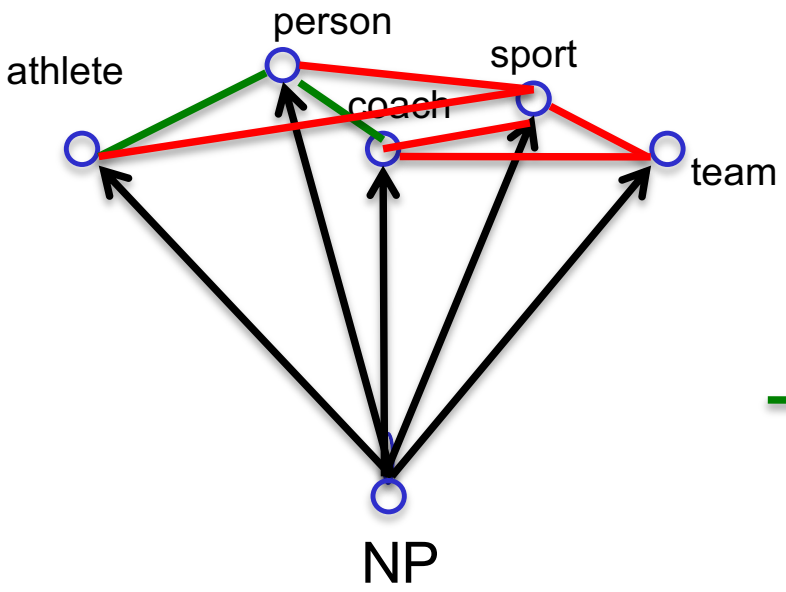
Then f_1, f_2 are PAC learnable from polynomial *unlabeled* data plus a weak initial predictor

Type 1 Coupling: Co-Training, Multi-View Learning

[Blum & Mitchell; 98]
[Dasgupta et al; 01]
[Balcan & Blum; 08]
[Ganchev et al., 08]
[Sridharan & Kakade, 08]
[Wang & Zhou, ICML10]



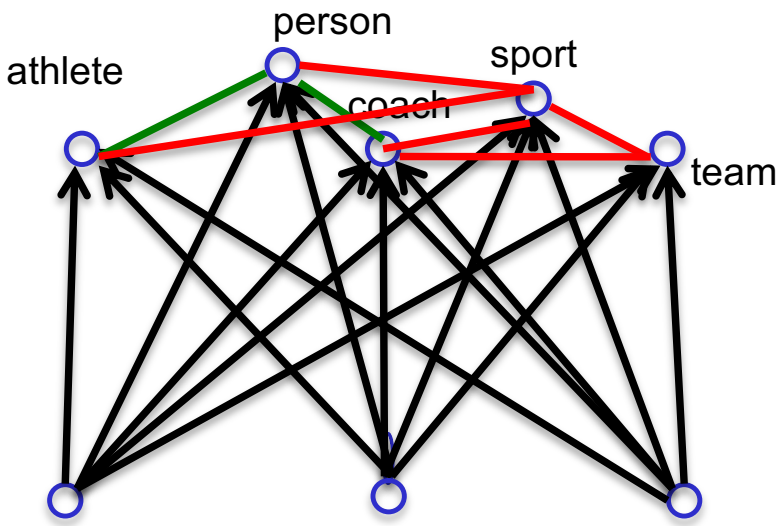
Type 2 Coupling: Multi-task, Structured Outputs



subset/superset
athlete(NP) → person(NP)

mutual exclusion
athlete(NP) → NOT sport(NP)
sport(NP) → NOT athlete(NP)

Multi-view, Multi-Task Coupling



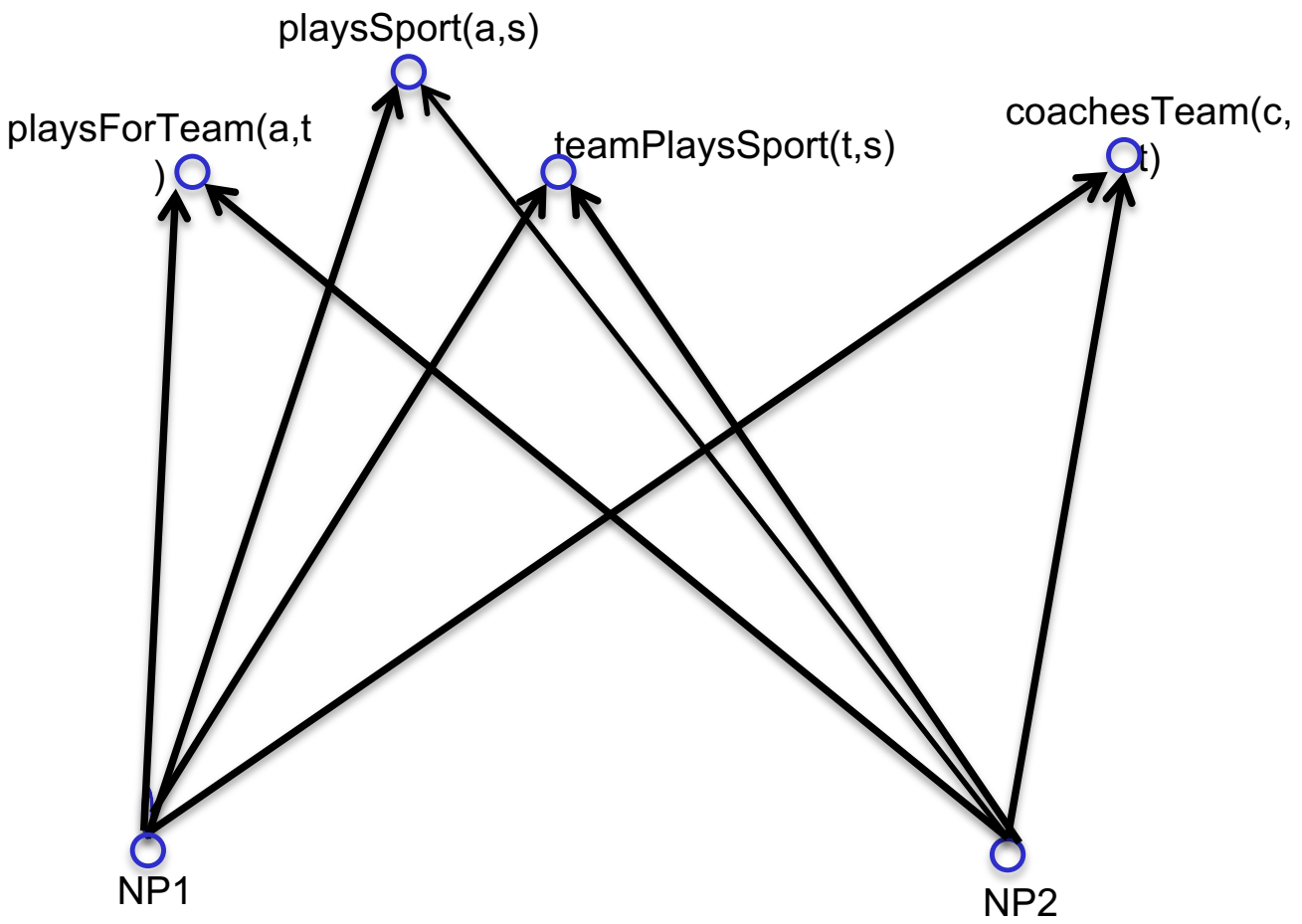
NP:

NP text context distribution

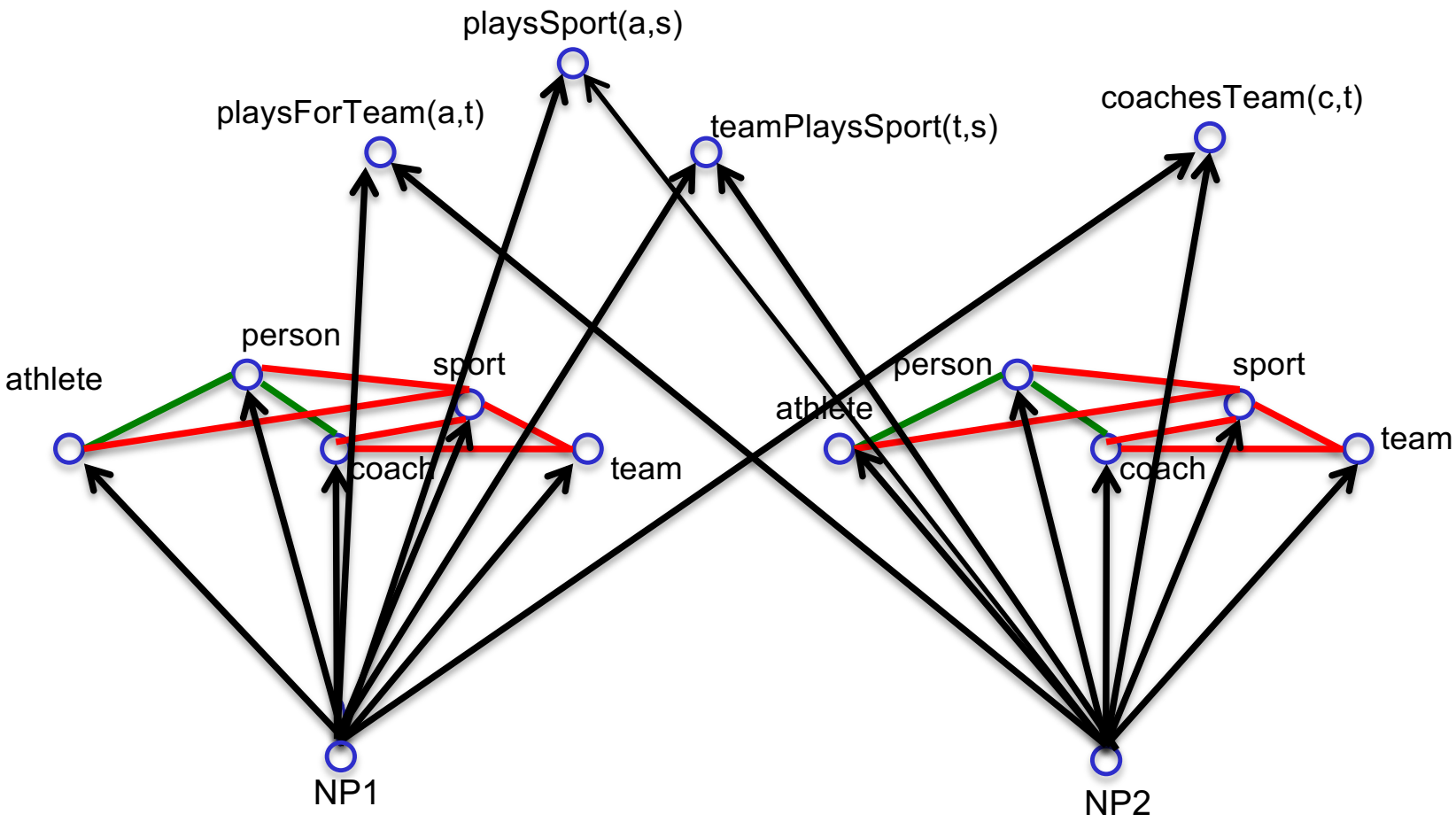
NP morphology

NP HTML contexts

Type 3 Coupling: Relations and Argument Types

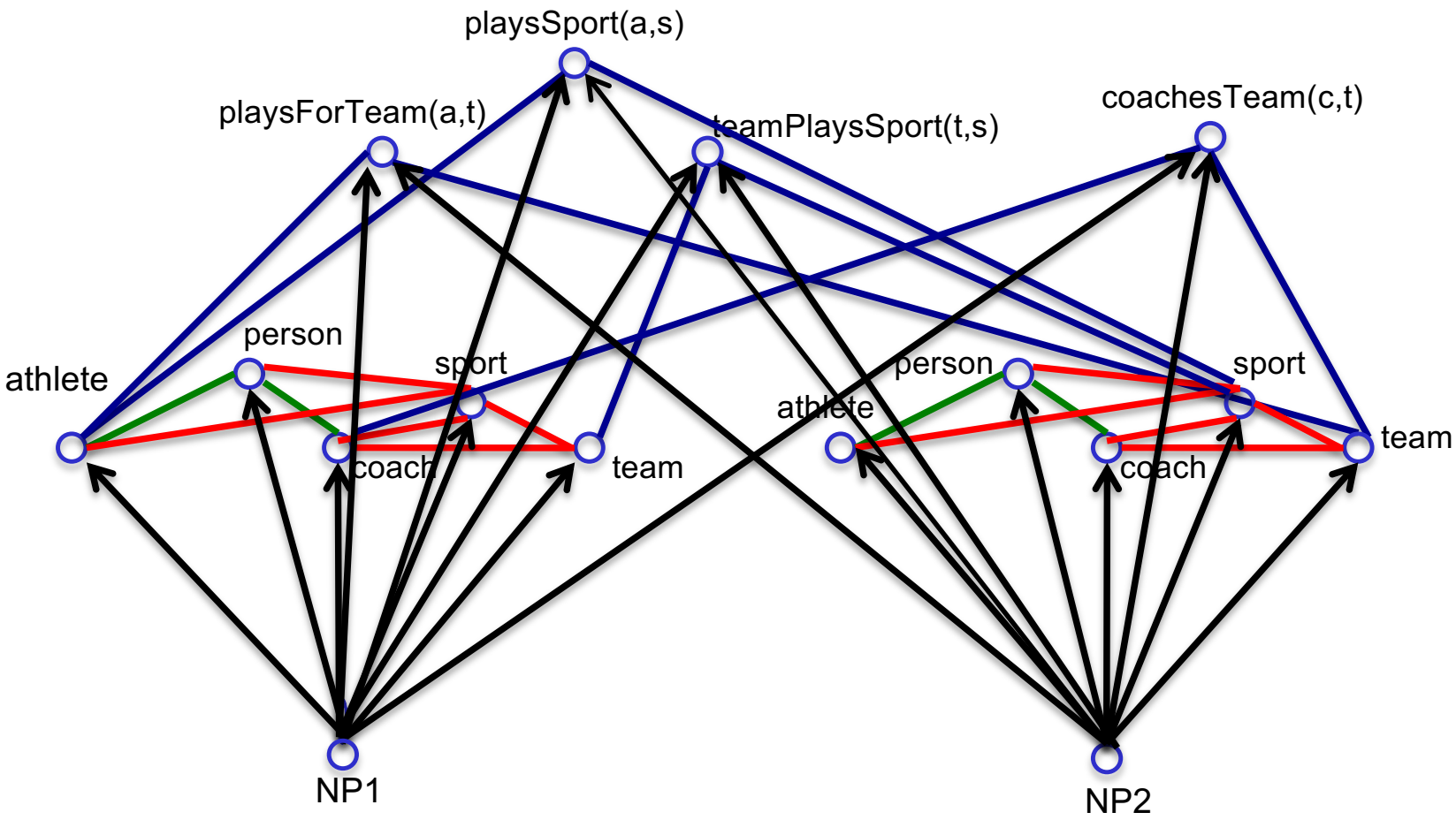


Type 3 Coupling: Relations and Argument Types



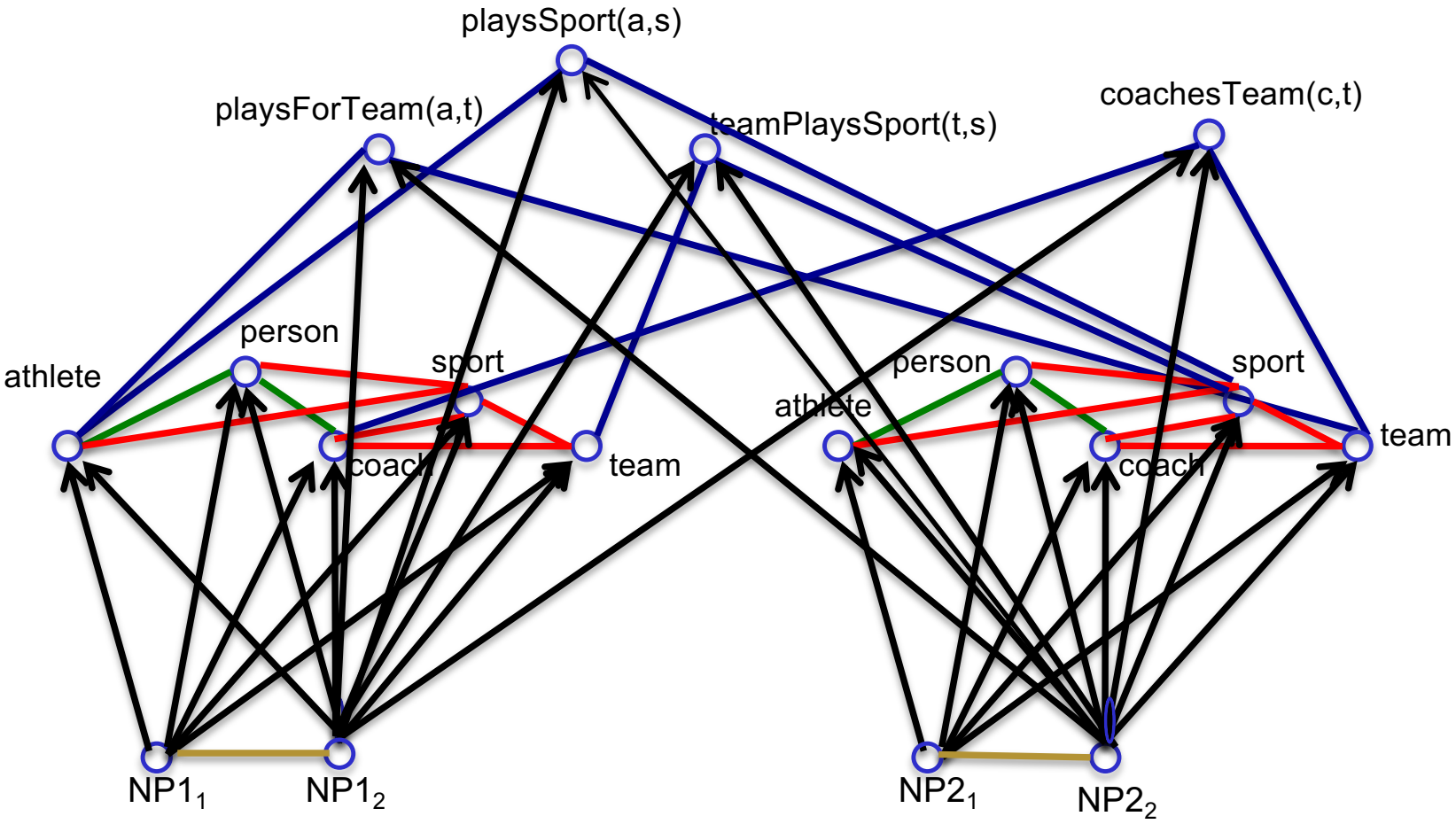
Type 3 Coupling: Relations and Argument Types

playsSport(NP1,NP2) → athlete(NP1), sport(NP2)



Type 3 Coupling: Relations and Argument Types

over 4000 coupled functions in NELL



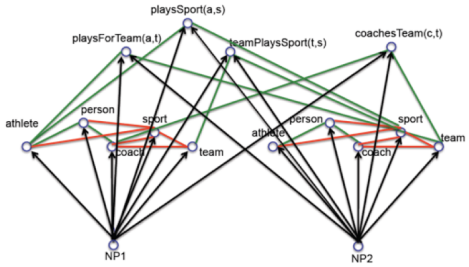
Q: What initial A structure allows A to learn from unlabeled data?

Ans: Couple the training of many functions
that capture overlapping information

Q: What architectures allow an agent to learn to learn?

i.e., where learning functions of type 1 **improves** the ability to learn functions of type 2

Learn new coupling constraints



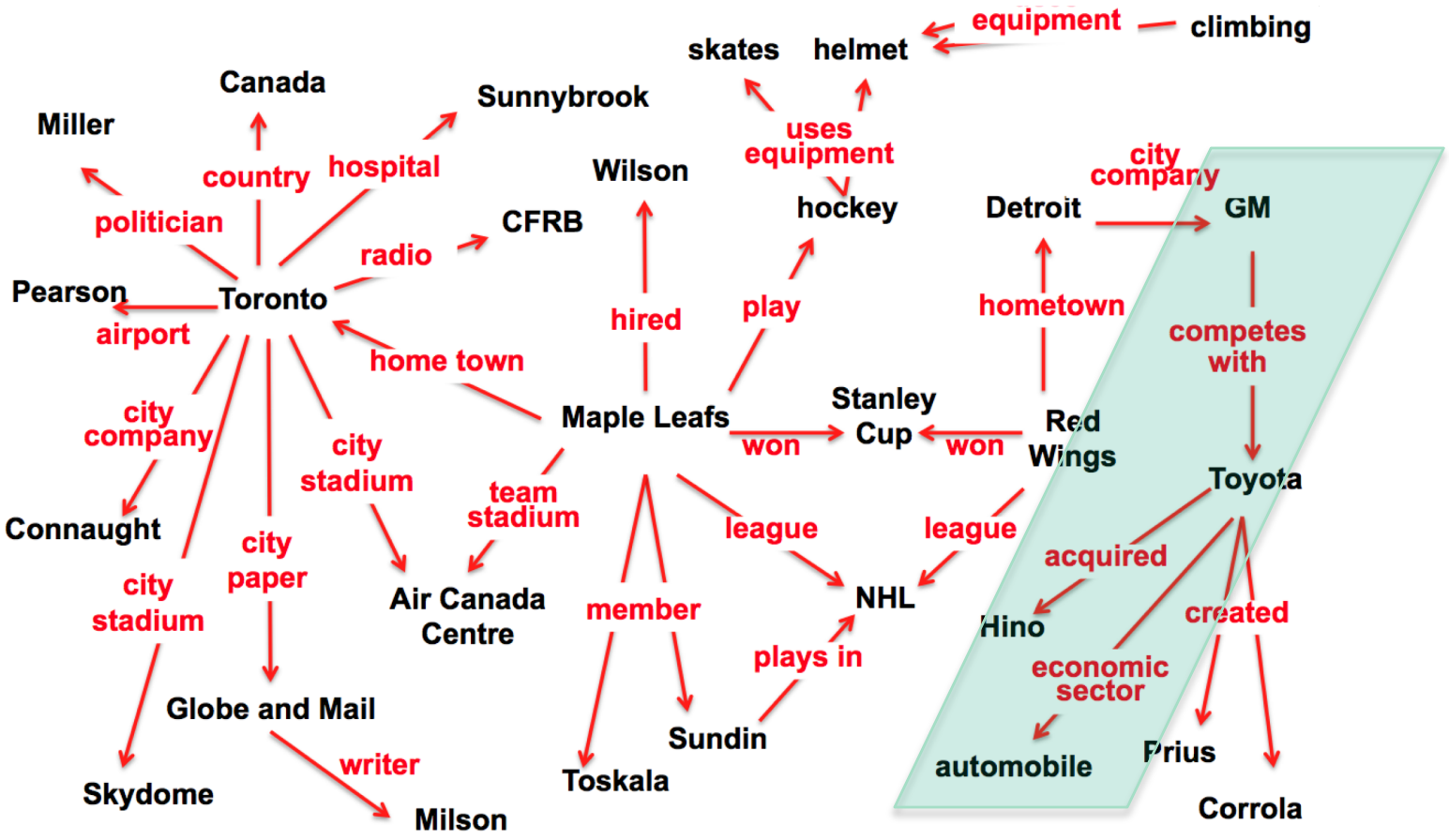
- first order, probabilistic horn clause constraints:

$$0.93\ \text{athletePlaysSport}(?x,?y) \leftarrow \text{athletePlaysForTeam}(?x,?z)\ \text{teamPlaysSport}(?z,?y)$$

  - learned from millions of beliefs in the knowledge base
  - connect previously uncoupled relation predicates
  - NELL has learned100,000s of such rules
  - uses PRA random-walk inference [Lao, Cohen, Gardner]

Learn inference rules

PRA: [Lao, Mitchell, Cohen, *EMNLP* 2011]

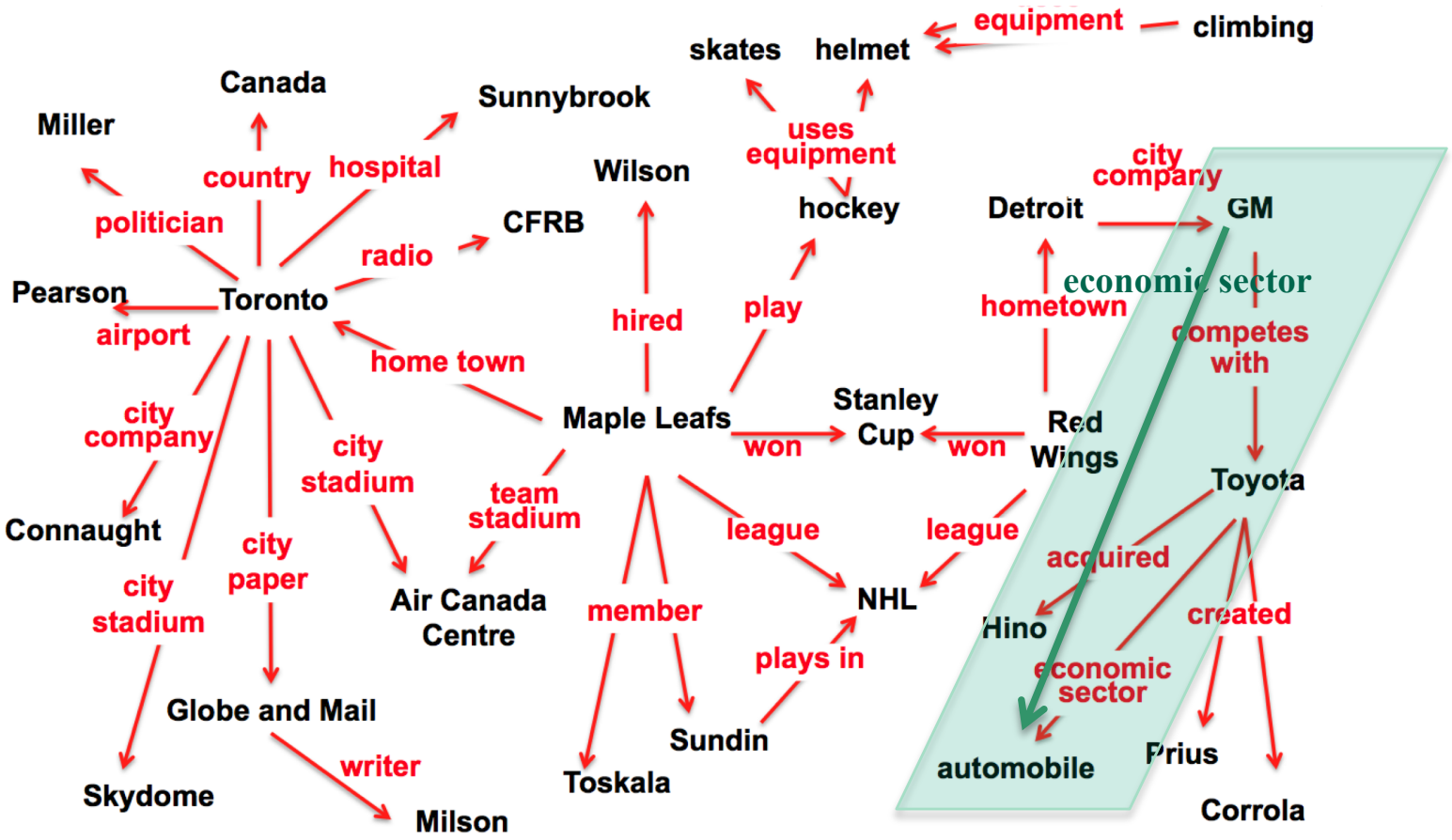


If: x1 — competes with (x1,x2) → x2 — economic sector (x2, x3) → x3

Then: economic sector (x1, x3) with probability 0.9

Learn inference rules

PRA: [Lao, Mitchell, Cohen, *EMNLP* 2011]

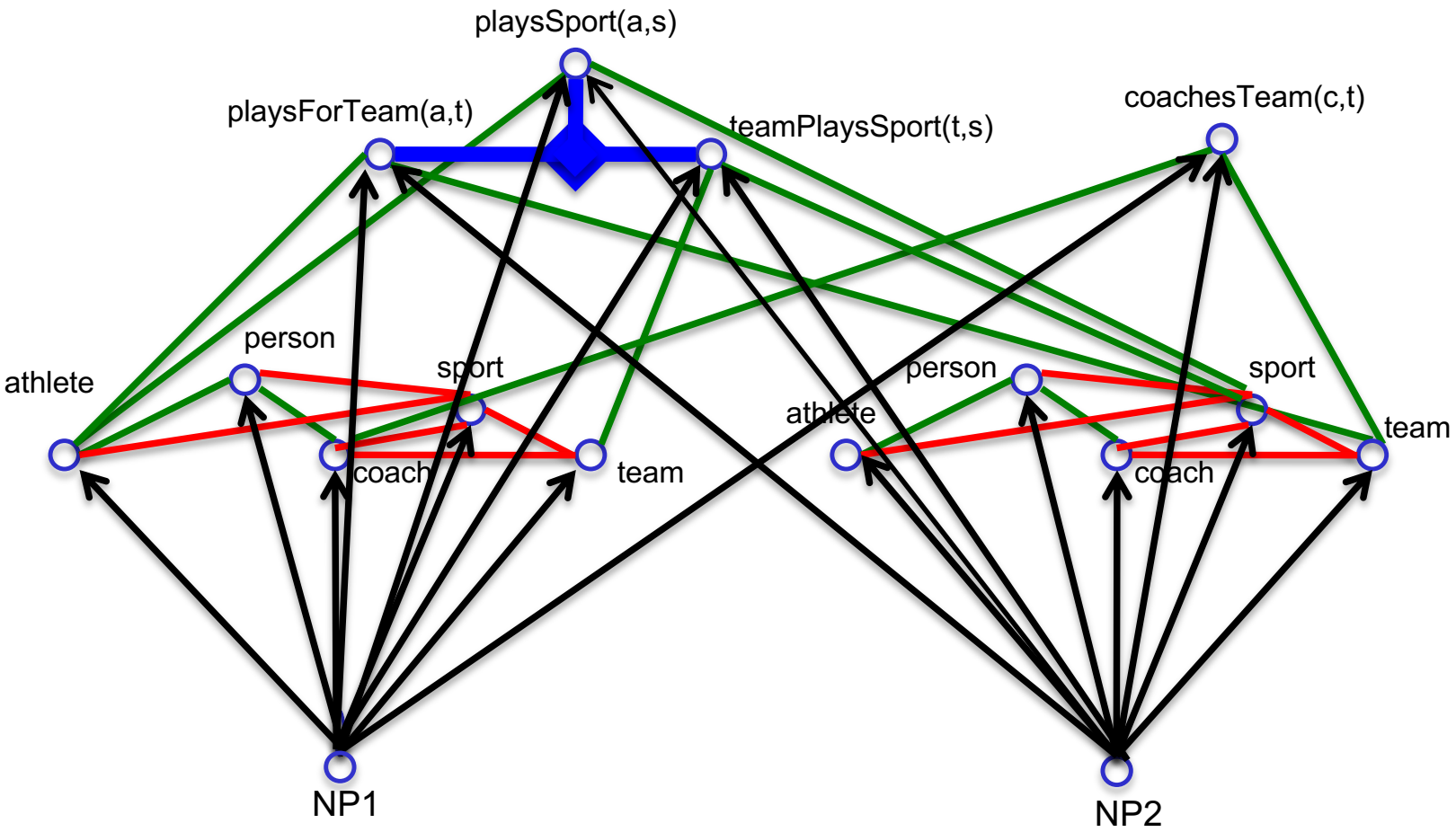


If: x1 — competes with (x1,x2) → x2 — economic sector (x2, x3) → x3

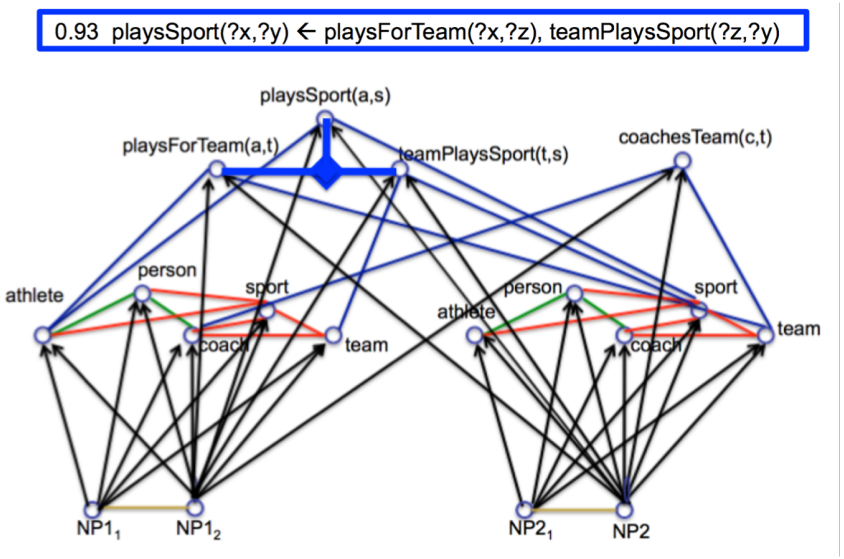
Then: economic sector (x1, x3) with probability 0.9

Learned Rules are New Coupling Constraints

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



Learned Rules are New Coupling Constraints

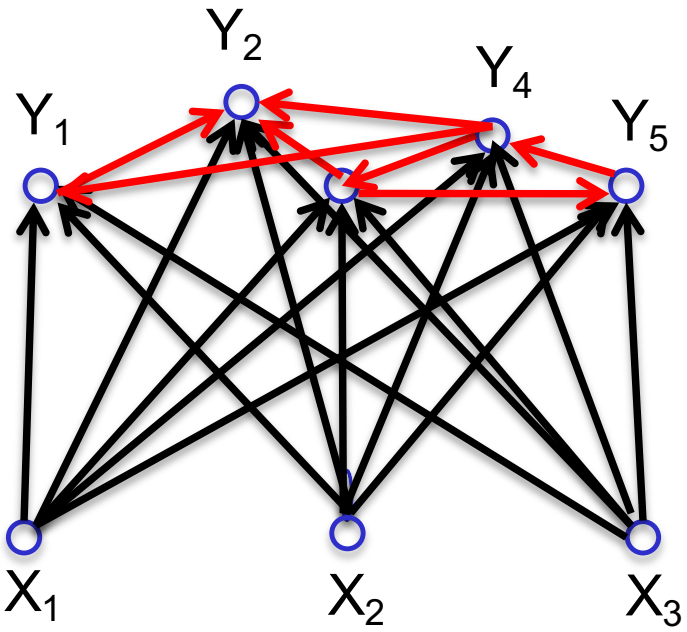


- Learning A makes one a better learner of B
- Learning B makes one a better learner of A

A = reading functions: `text` → `beliefs`

B = Horn clause rules: `beliefs` → `beliefs`

Q: Can we prove conditions under which learning both type 1 and type 2 functions, from the same data, improves ability to learn type 1 functions?



Type 1 functions: $f_{ik}: X_i \rightarrow Y_k$
Type 2 functions: $g_{nm}: Y_n \rightarrow Y_m$

Can we find conditions under which we lower the unlabeled sample complexity for learning all f_{ik} functions, by adding the tasks of also learning the g_{nm} functions?

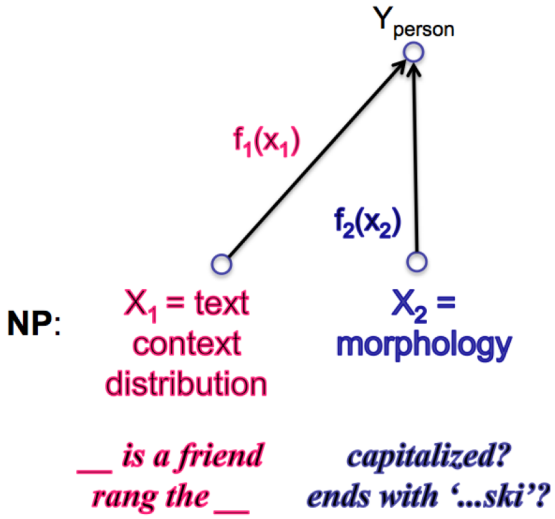
Conjecture: yes

Self-Reflection

Q: what architectures allow agent to estimate accuracy of learned functions, given only *unlabeled data*?

Self-Reflection

Q: what architectures allow agent to estimate accuracy of learned functions, given only *unlabeled data*?



[Platanios, Blum, Mitchell]

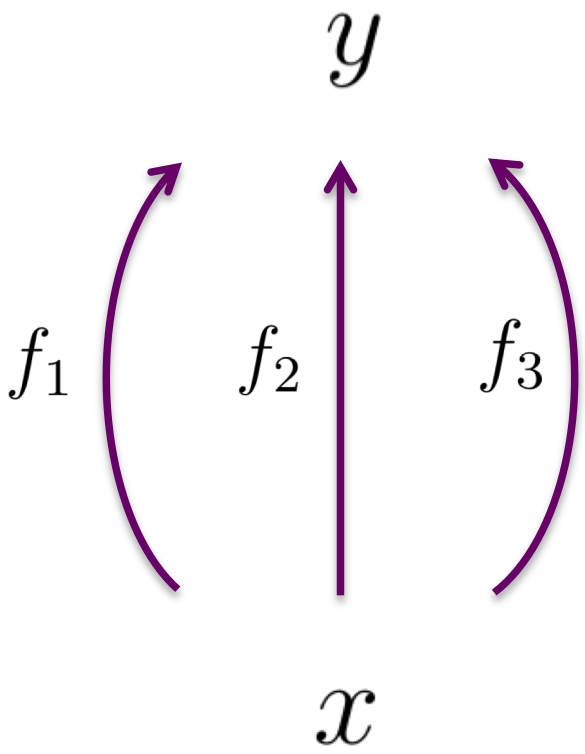
Problem setting:

- have N different estimates $f_1, \ldots f_N$ of target function f^*

$$f^* : X \rightarrow Y; \quad Y \in \{0, 1\}$$

Goal:

- estimate accuracy of each of $f_1, \ldots f_N$ from **unlabeled** data



Example:

y = NELL category "hotel"

f_i = classifier based on ith view of x

x = noun phrase

[Platanios, Blum, Mitchell]

Problem setting:

- have N different estimates $f_1, \ldots f_N$ of target function f^*

- define *agreement* between f_i, f_j : $a_{ij} \equiv P_x(f_i(x) = f_j(x))$

Problem setting:

- have N different estimates $f_1, \ldots f_N$ of target function f^*

- define *agreement* between f_i, f_j : $a_{ij} \equiv P_x(f_i(x) = f_j(x))$

Note agreement can be estimated with unlabeled data

$$a_{ij} = \text{Pr[neither makes error]} + \text{Pr[both make error]}$$

$$a_{ij} = 1 - e_i - e_j + 2e_{ij}$$

a_{ij}	e_i	e_j	$2e_{ij}$
prob. f_i and f_j agree	prob. f_i error	prob. f_j error	prob. f_i and f_j simultaneous error

Estimating Error from Unlabeled Data

1. IF f_1, f_2, f_3 make <u>independent</u> errors,

   then $a_{ij} = 1 - e_i - e_j + 2e_{ij}$

   becomes $a_{ij} = 1 - e_i - e_j + 2e_i e_j$

prob. f_i and f_j simultaneous error

Estimating Error from Unlabeled Data

prob. f_i and f_j simultaneous error

1. IF f_1, f_2, f_3 make <u>independent</u> errors,

   then $$a_{ij} = 1 - e_i - e_j + 2e_{ij}$$

   becomes $$a_{ij} = 1 - e_i - e_j + 2e_i e_j$$

If errors independent, and $e_1 < 0.5$, $e_2 < 0.5$, then

- use unlabeled data to estimate a_{12}, a_{13}, a_{23}. Solve for error rates

$$a_{12} = 1 - e_1 - e_2 + 2e_1 e_2$$

$$a_{13} = 1 - e_1 - e_3 + 2e_1 e_3$$

$$a_{23} = 1 - e_2 - e_3 + 2e_2 e_3$$

Estimating Error from Unlabeled Data

1. IF f_1, f_2, f_3 make indep. errors, accuracies > 0.5

   then $$a_{ij} = 1 - e_i - e_j + 2e_{ij}$$

   becomes $$a_{ij} = 1 - e_i - e_j + 2e_i e_j$$

2. but what if errors **not** independent?

Estimating Error from Unlabeled Data

1. IF f_1, f_2, f_3 make indep. errors, accuracies > 0.5

   then $$a_{ij} = 1 - e_i - e_j + 2e_{ij}$$

   becomes $$a_{ij} = 1 - e_i - e_j + 2e_i e_j$$

2. but if errors **not** independent, add prior:
   the more independent, the more probable

$$\min \sum_{i,j} (e_{ij} - e_i e_j)^2$$

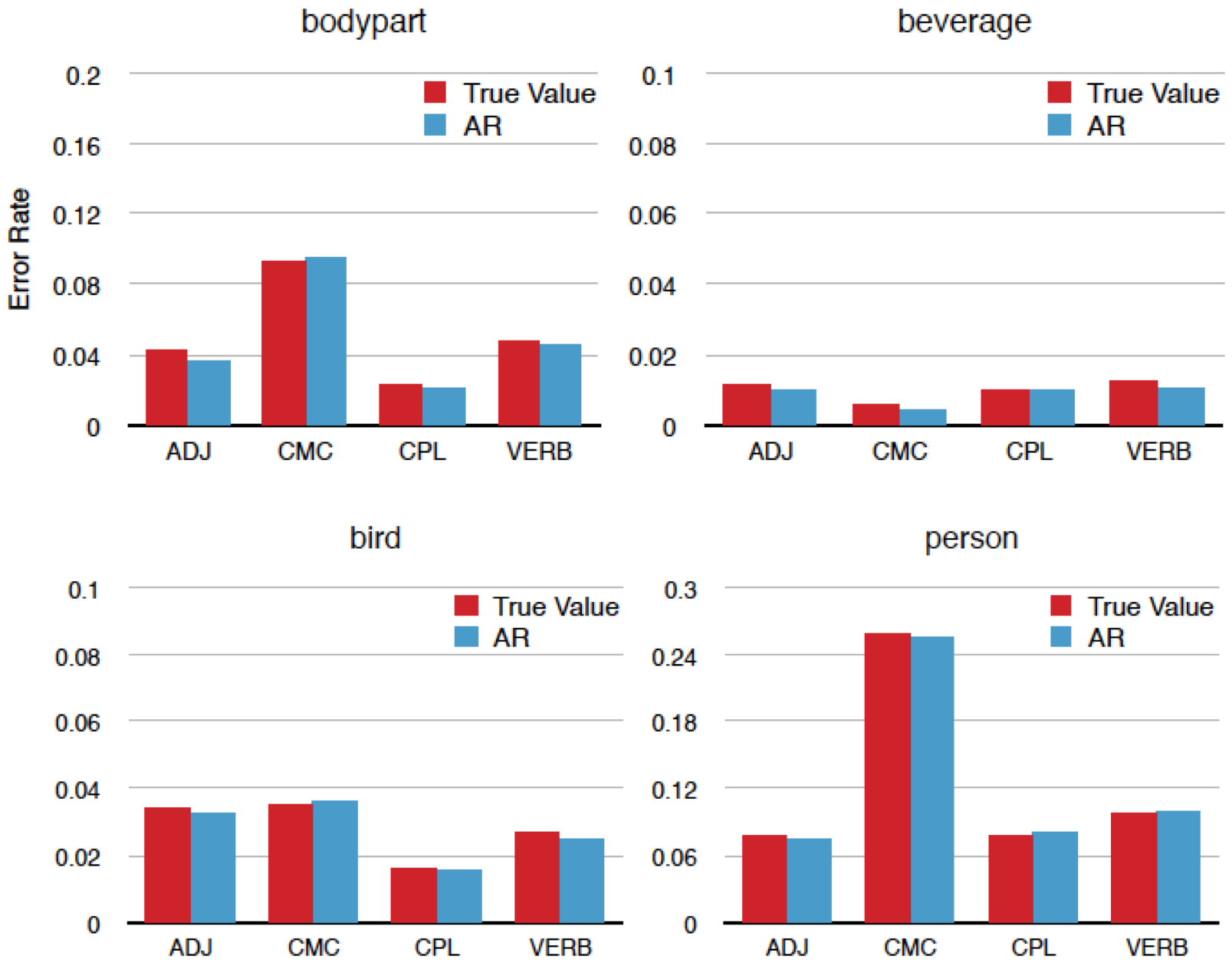
such that

$$(\forall i, j)\ a_{ij} = 1 - e_i - e_j + 2e_{ij}$$

True error (red), estimated error (blue)

[Platanios et al., 2014]

NELL classifiers:



Self-Reflection

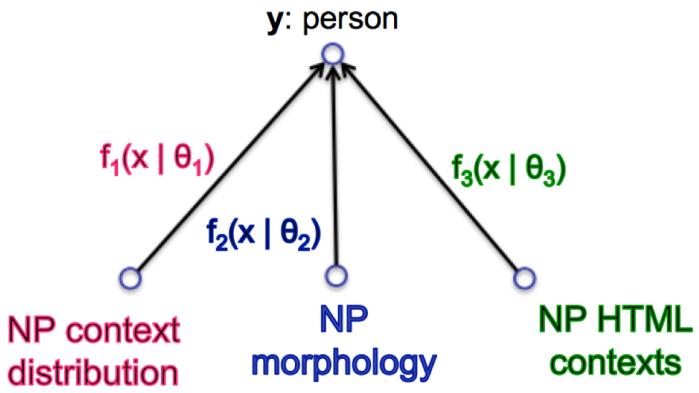
Q: what architectures allow agent to estimate accuracy of its learned functions, given only *unlabeled data*?

Ans: Again, architectures that have many functions, capturing overlapping information

Multiview setting

Given functions f_i: $X_i \rightarrow \{0,1\}$ that

- make independent errors
- are better than chance



If you have at least **2** such functions

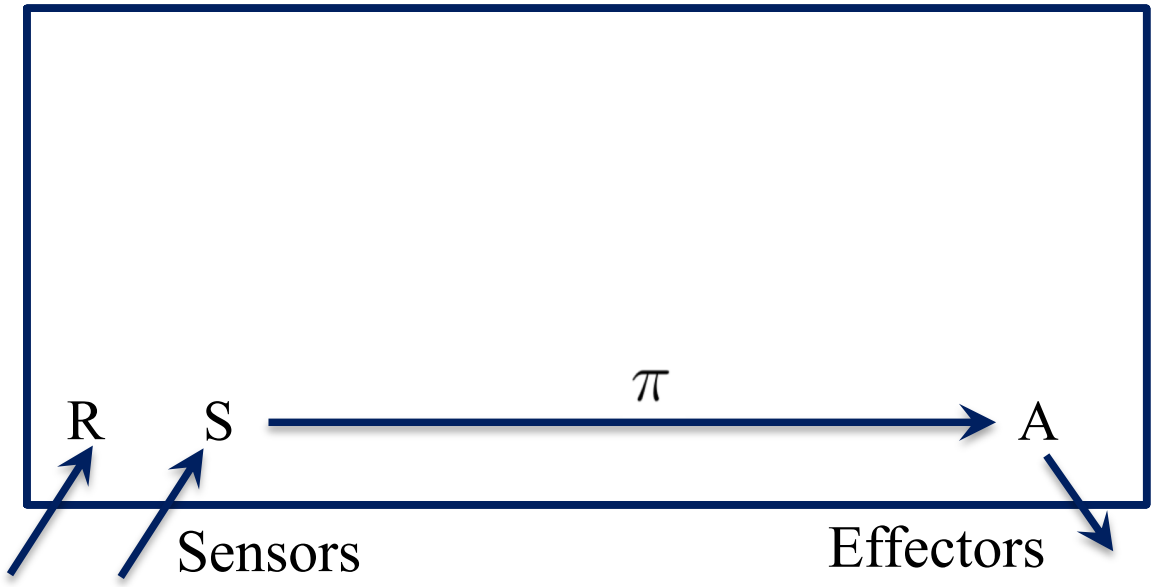
- they can be PAC learned by co-training them to agree over unlabeled data [Blum & Mitchell, 1998]

If you have at least **3** such functions

- their accuracy can be calculated from agreement rates over unlabeled data [Platanios et al., 2014]

Q: Is accuracy estimation strictly harder than learning?

Reinforcement Learning



Setting: States *S*, Actions *A*

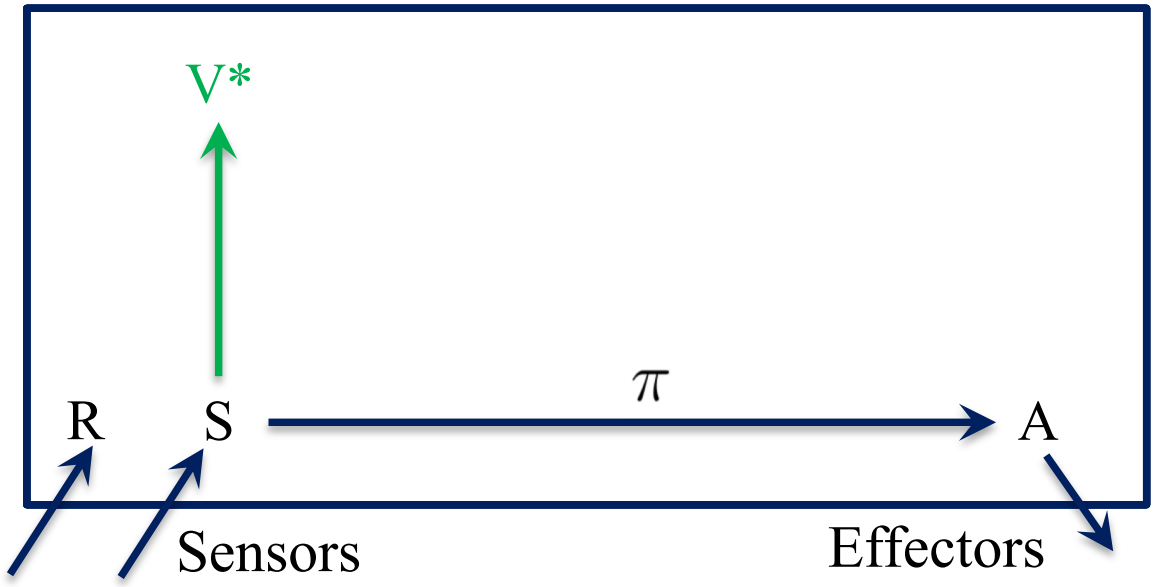
Learn a policy $\pi : S \rightarrow A$

that optimizes sum of rewards discounted over time:

$$\pi^* = \arg\max_{\pi} \ \Sigma_{t=0}^{\infty} \ \gamma^t R(s_t, \pi(s_t))$$

Learn:

$$\pi : S \rightarrow A$$



Setting: States *S*, Actions *A*

Learn a policy $\pi : S \rightarrow A$

that optimizes sum of rewards discounted over time:

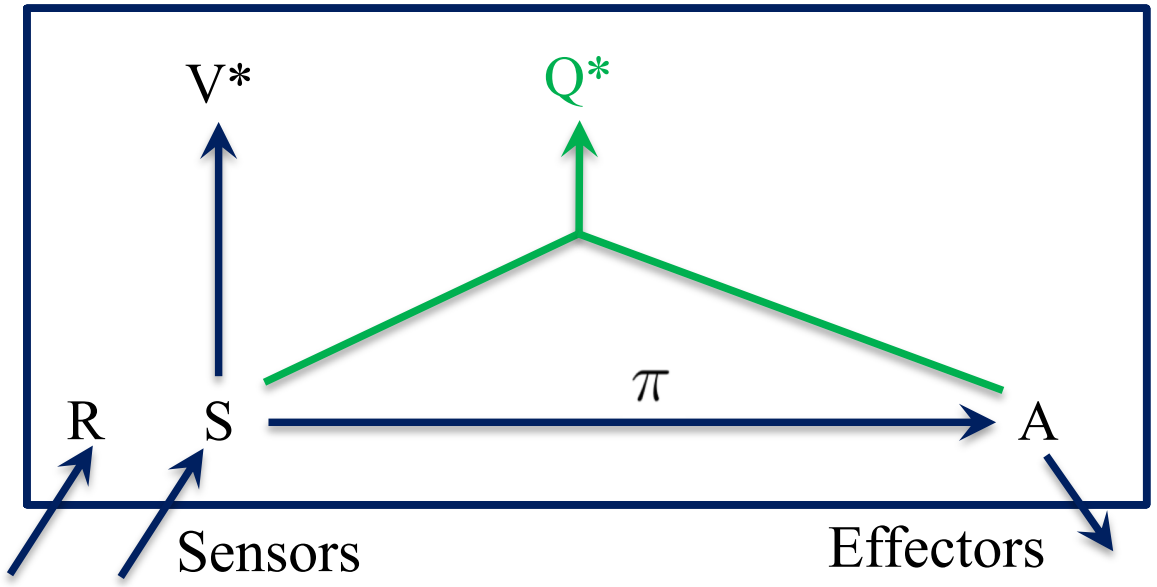
$$\pi^* = \arg\max_{\pi} \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))$$

Learn:

$$\pi : S \rightarrow A$$

$$V^* : S \rightarrow \Re$$

where $V^*(s) = \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$



Setting: States S, Actions A

Learn a policy $\pi : S \rightarrow A$

that optimizes sum of rewards discounted over time:

$$\pi^* = \arg\max_{\pi} \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))$$

Learn:

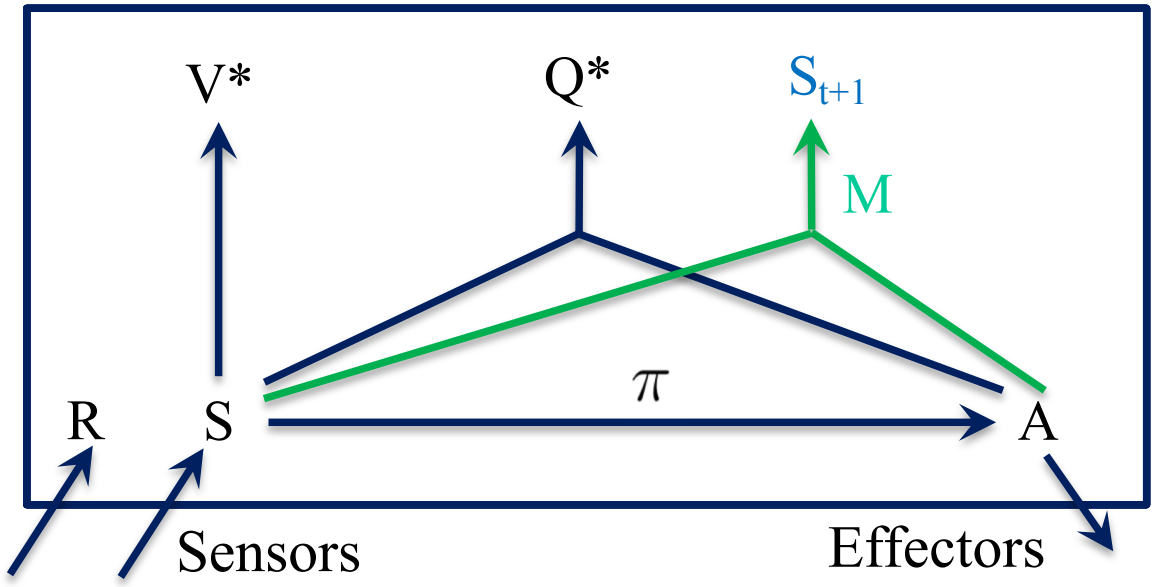
$$\pi : S \rightarrow A$$

$$V^* : S \rightarrow \Re$$

where $V^*(s) = \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$

$$Q^* : S \times A \rightarrow \Re$$

$$Q^*(s,a) = R(s,a) + \Sigma_{t=1}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$$



Setting: States *S*, Actions *A*

Learn a policy $\pi : S \rightarrow A$

that optimizes sum of rewards discounted over time:

$$\pi^* = \arg\max_{\pi} \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))$$

Learn:

$$\pi : S \rightarrow A$$

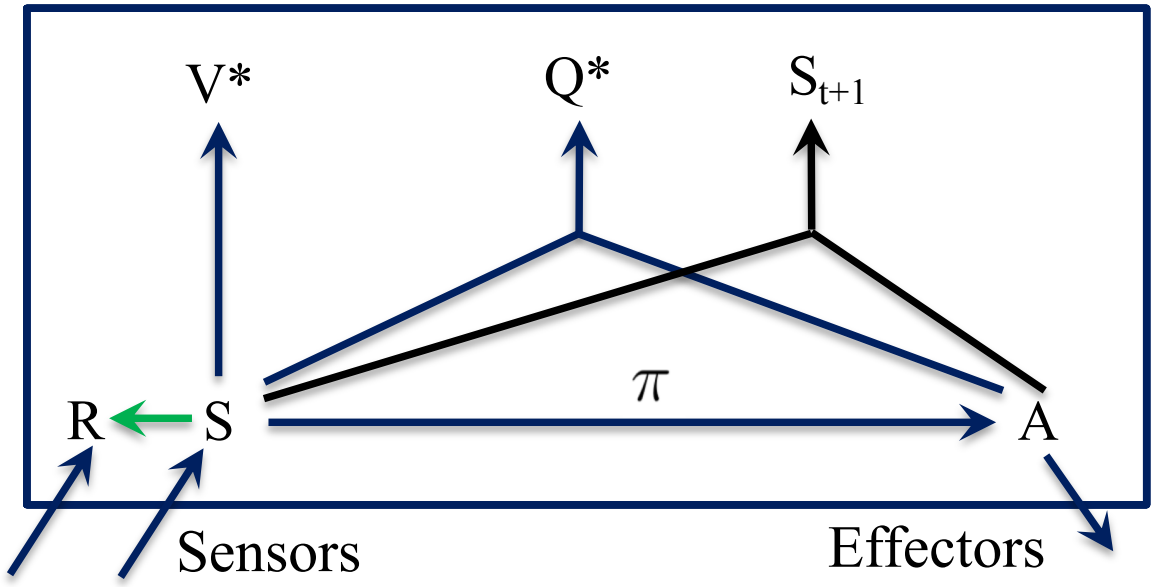
$$V^* : S \rightarrow \Re$$

where $V^*(s) = \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$

$$Q^* : S \times A \rightarrow \Re$$

$$Q^*(s,a) = R(s,a) + \Sigma_{t=1}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$$

$$M : S_t \times A \rightarrow S_{t+1}$$



Setting: States *S*, Actions *A*

Learn a policy $\pi : S \rightarrow A$

that optimizes sum of rewards discounted over time:

$$\pi^* = \arg\max_{\pi} \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))$$

Learn:

$$\pi : S \rightarrow A$$

$$V^* : S \rightarrow \Re$$

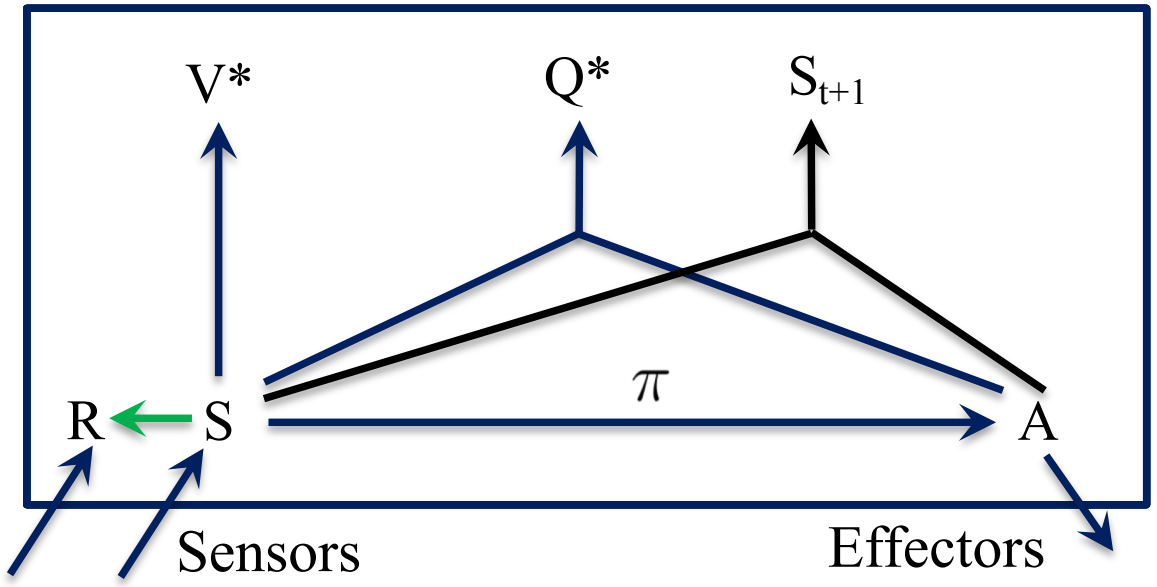
where $V^*(s) = \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$

$$Q^* : S \times A \rightarrow \Re$$

$$Q^*(s,a) = R(s,a) + \Sigma_{t=1}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$$

$$M : S_t \times A \rightarrow S_{t+1}$$

$$R : S \times A \rightarrow \Re$$



Note these functions inter-related!

→ Coupled training from unlabeled data

- Actor-critic methods learn V* and π jointly
- Coupling constraints among other functions as well, e.g.,

$$V^*(s) = \max_a Q^*(s,a)$$

Learn:

$$\pi : S \rightarrow A$$

$$V^* : S \rightarrow \Re$$

where $V^*(s) = \Sigma_{t=0}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$

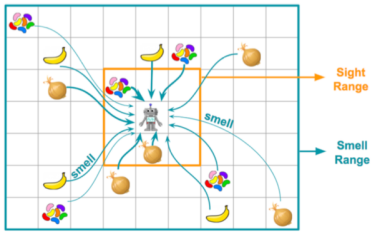
$$Q^* : S \times A \rightarrow \Re$$

$$Q^*(s,a) = R(s,a) + \Sigma_{t=1}^{\infty} \gamma^t R(s_t, \pi^*(s_t))$$

$$M : S_t \times A \rightarrow S_{t+1}$$

$$R : S \times A \rightarrow \Re$$

Coupled training of V*(s) and Q*(s,a)



Represent V(s), Q(s,a) as two neural nets, train at each step to minimize sq error violation of coupling constraint

$$V^*(s) = \max_a Q^*(s,a)$$

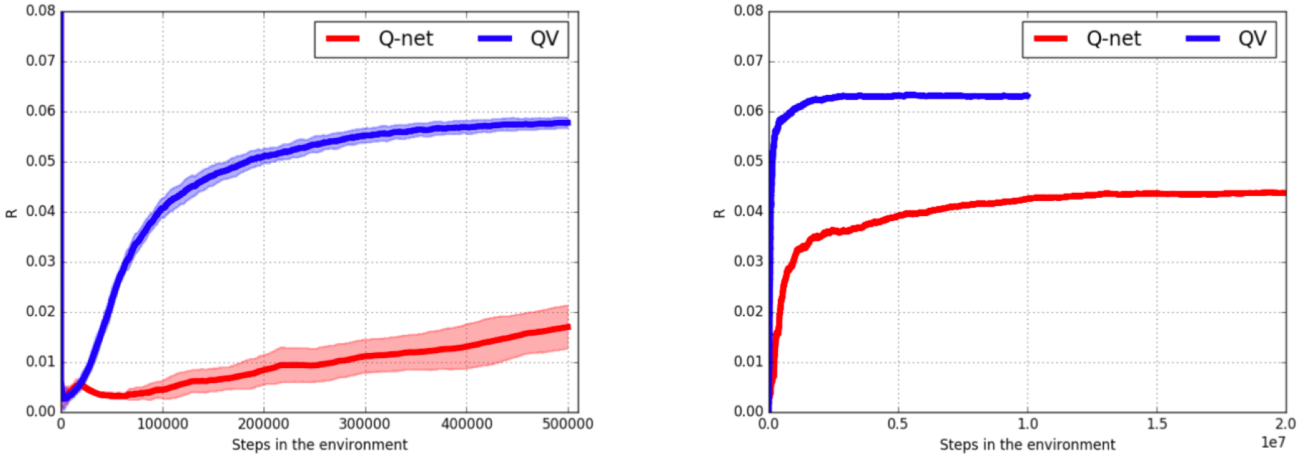


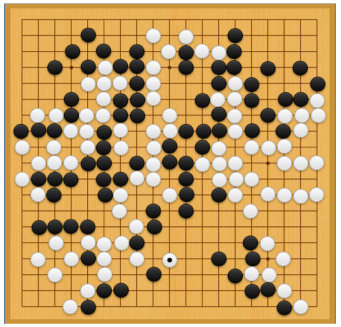
Figure 5: Training performance of coupled training agent 'QV' with respect to the performance of baseline agent 'Q-net'. **(left)** 'QV' agent vs baseline agent performance over 10 runs for 500,000 steps in environment. **(right)** Asymptotic performance of 'QV' agent vs baseline agent for a single run. 'QV' agent was run for 10 million steps, baseline agent was run for 20 million steps in the environment.

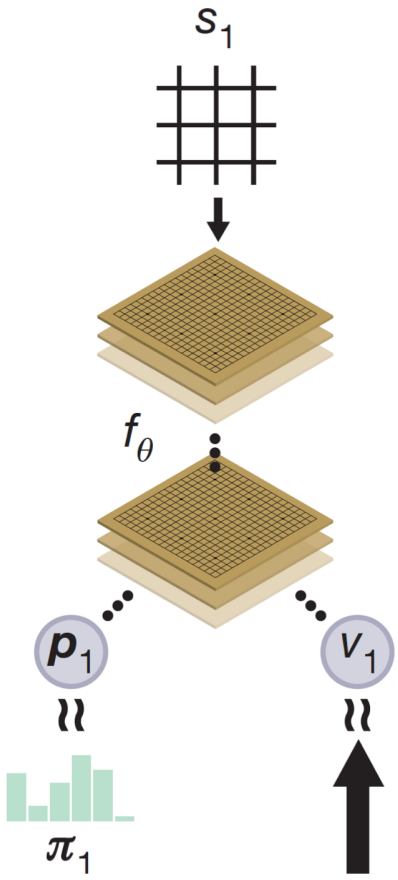
[Ozutemiz & Bhotika, 2018, class project]

(based on Deep Q Learning w/experience replay [Mnih, et al. 2015])

Alpha Go Zero coupled training of. $\pi : S \to A, \quad V^* : S \to \Re$

Coupling by shared neural network to learn shared state representation





Reinforcement learning – conclusions

- Good fit to deep networks
- Coupled unsupervised training of multiple functions
- Couple either
  - Through shared representation (e.g., Alpha Go Zero)
  - Through explicit coupling of independently represented functions
- Self-supervised data available for some functions

$$M : S_t \times A \rightarrow S_{t+1} \qquad R : S \times A \rightarrow \Re$$

- Conjecture: further improvements possible by adding yet more inter-related functions, and coupling their training …

$$A(s,a) \equiv V^*(a(s)) - V^*(s)$$
$$D(s,a) \equiv M(s,a) - s$$

Reinforcement learning – many extensions

- Experience replay
- Imitation learning
- Hierarchical actions
- Reward shaping
- Curiosity-driven learning
- …

Self-Reflection

Q: How can we architect a never-ending learning agent so that it can notice every learning need, and address it?

Self-Reflection

Q: How can we architect a never-ending learning agent so that it can notice every learning need, and address it?

SOAR: A Case Study

Soar: An architecture for general intelligence JE Laird, A Newell, PS Rosenbloom - Artificial intelligence, 1987.

The Soar cognitive architecture MIT Press, JE Laird - 2012

SOAR

[Laird, Newell, Rosenbloom, 1987]

[Laird, 2012].

Design philosophy:

- Self-reflection that can detect every possible shortcoming (called *impasse*) of the agent
- There are only four types of impasses
- Every instance of an impasse can be solved using a (potentially expensive) built in method
- Every solved impasse results in learning an if-then rule that will pre-empt that impasse in the future (and ones like it)

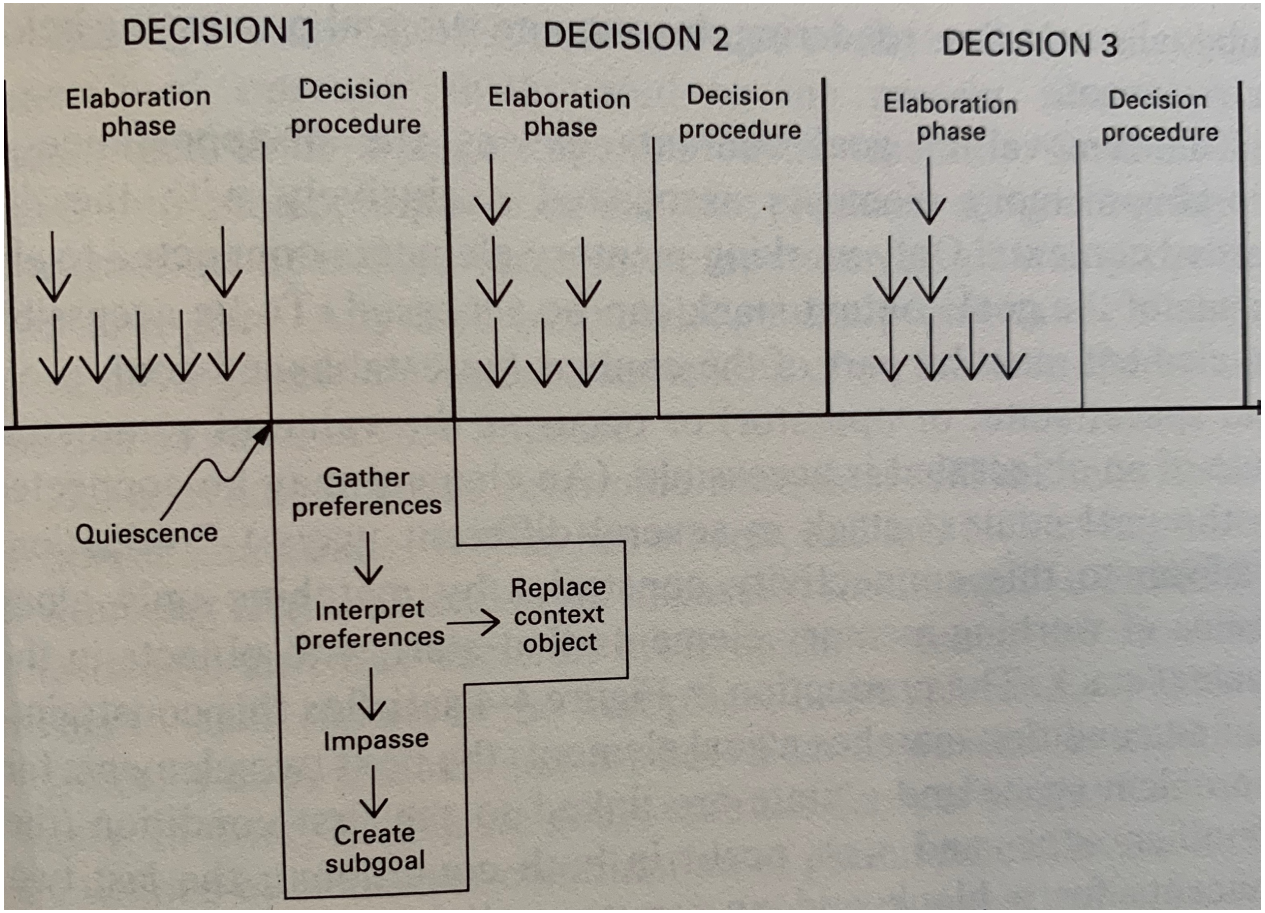
→ Every shortcoming will be noticed by the agent, and will result in learning to avoid it

SOAR

Key design elements:

- *Every* problem is treated as a search problem
- Self-reflection mechanism detects *every possible difficulty* in solving search problems (called *impasses*).

SOAR Decision Cycle

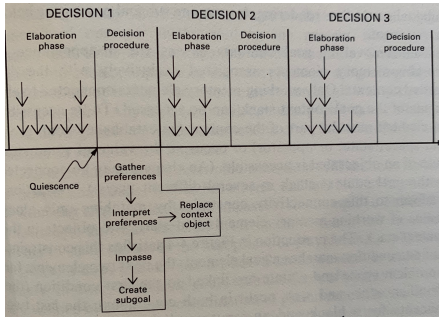


SOAR chooses

- Problem space
- Search state
- Operator

[Newell 1990]

SOAR



Key design elements:

- *Every* problem is treated as a search problem
- Self-reflection mechanism detects *every possible difficulty* in solving search problems (called *impasses*). Four types:
  - *Tie impasse* : among potential next steps, no obvious “best”
  - *No-change impasse* : no available next steps
  - *Reject impasse* : only available step is to reject options
  - *Conflict impasse* : incompatible recommendations for next step
- When impasse detected, architecture formulates the problem of resolving it, as a new search problem (in a different search space)
- Initial architecture seeded with weak search methods to solve all four impasses
- After resolving an impasse, SOAR creates a new rule that will pre-empt this (and similar) impasses in the future

SOAR - Example

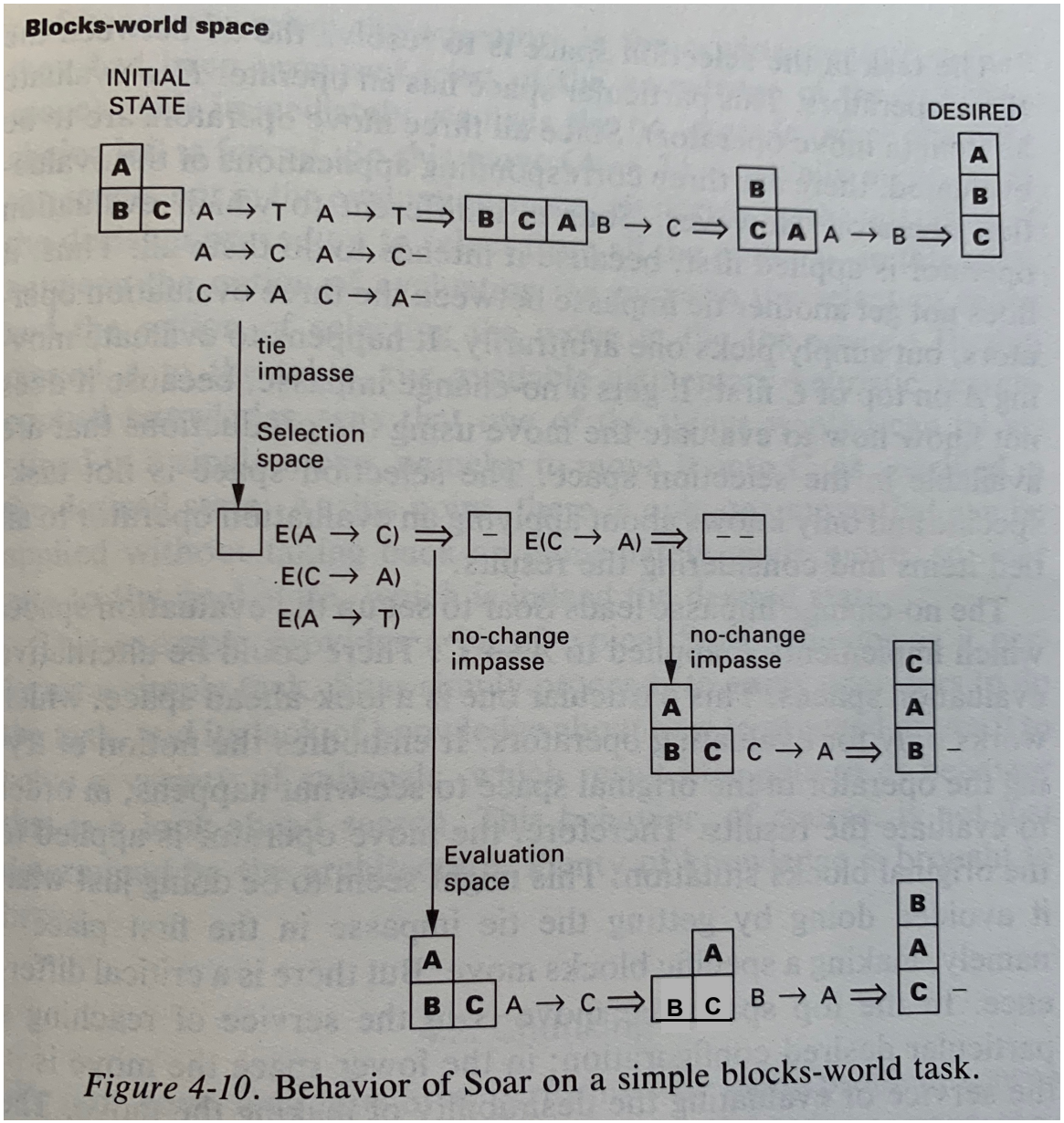
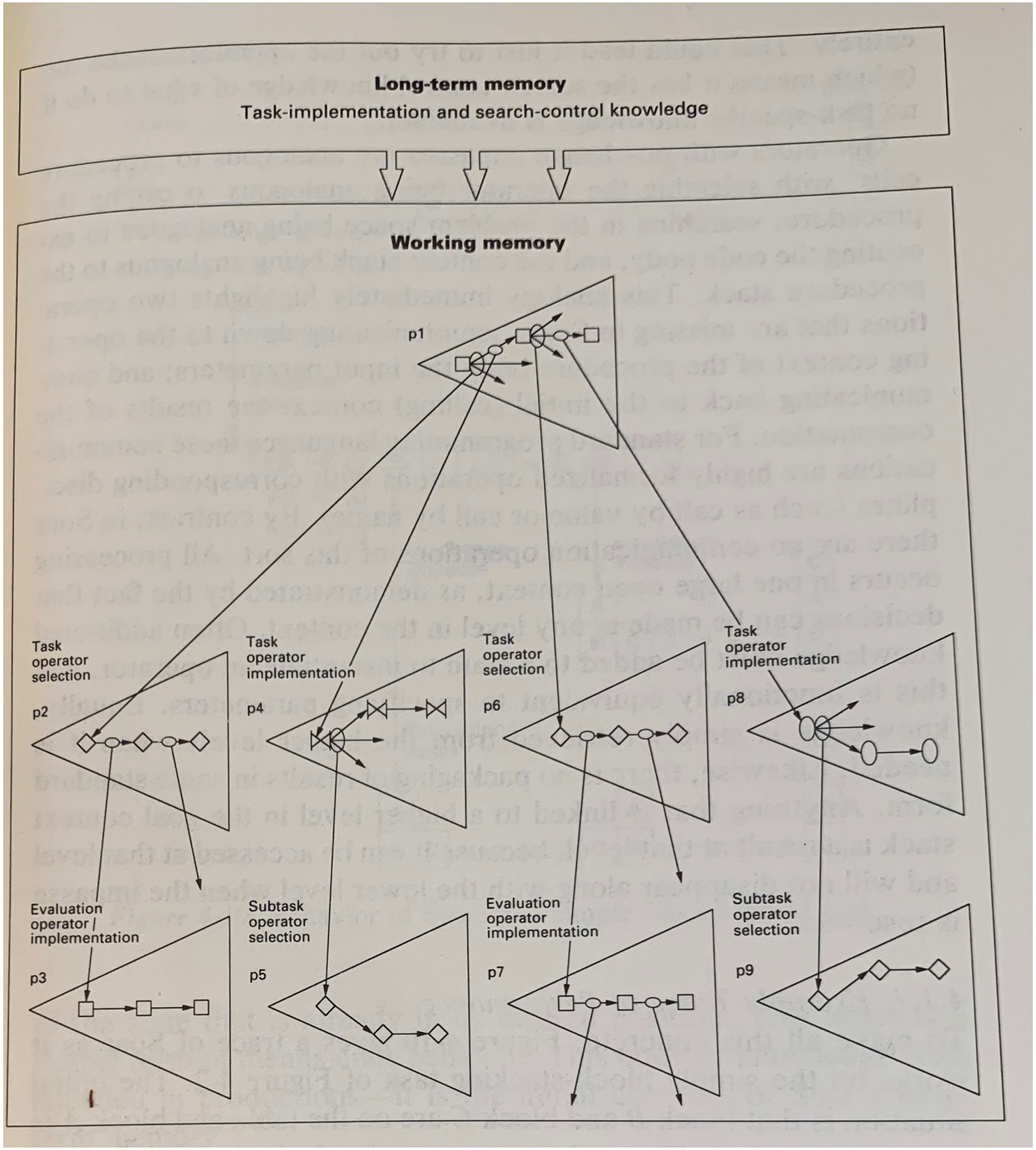


Figure 4-10. Behavior of Soar on a simple blocks-world task.

[Newell 1990]

SOAR



[Newell 1990]

SOAR

Lessons:

- Elegant architecture with *complete* self-reflection and learning
  - *Complete* = *every* need for learning noticed and addressed
- Built on a canonical representation of problem-solving as search

Then why didn't it solve the AI problem?

- It worked well for search problems with fully known actions and goal states, but…
- We lack accurate search operators for real robot actions
- Perception is hard to frame as search with a goal state
- Even for chess, didn't fully handle scaling up

Nevertheless: SOAR-TECH

COMPANY FOCUS

At SoarTech, our focus is in the development of intelligent software that reasons like humans do, to automate complex tasks, simplify human-machine interactions, or model human behaviors. Our philosophy is three-fold: to be an augmentation to, not a replacement of, the human; to think "top-down, not bottom-up;" and to be transparent so that decisions and processing are communicated to the human and in human-like terms.

WHERE WE SOAR

AUTONOMOUS PLATFORMS

Robotic intelligence frees humans from being tethered to a robot – and makes the robot a true teammate instead of a piece of equipment.

CYBERSPACE

SoarTech applies AI techniques and cognitive science towards automating cyberspace operations, command and control, and training.

DECISION SUPPORT

The most complex thing humans do is decide. At SoarTech, we build intelligent software to mimic this process, thereby enhancing a human's decision making.

INTELLIGENT TRAINING

SoarTech is a world leader in the development of intelligent behavior models to support simulation based and immersive training applications.

Are you ready to help define the future? JOIN US

Never-Ending Learning

ICML 2019 Tutorial: Part II

Tom Mitchell Partha Talukdar

https://sites.google.com/site/neltutorialicml19/

Research Issues

- Continual Learning and Catastrophic Forgetting
- (External) Knowledge and Reasoning
- Representation Learning
- Self Reflection
- Curriculum Learning

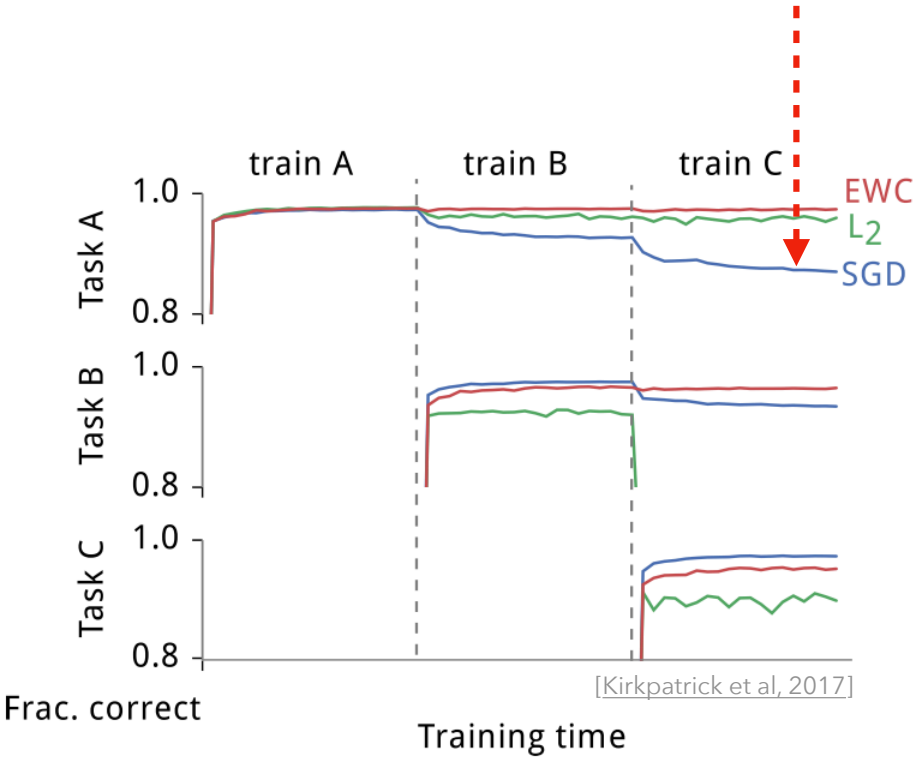
Continual Learning (CL)

- Tasks arrive sequentially: T_1, T_2, T_3, …
- One approach: Multitask Learning (MTL) over all tasks so far
  - Effective but impractical: need to store data from all previous tasks and replayed for each new task
- What we need: learn new task well
  - without having to store and replay data from old tasks
  - without losing performance in old tasks: *catastrophic forgetting* (next)

Catastrophic Forgetting (CF)

[McCloskey and Cohen, 1989]

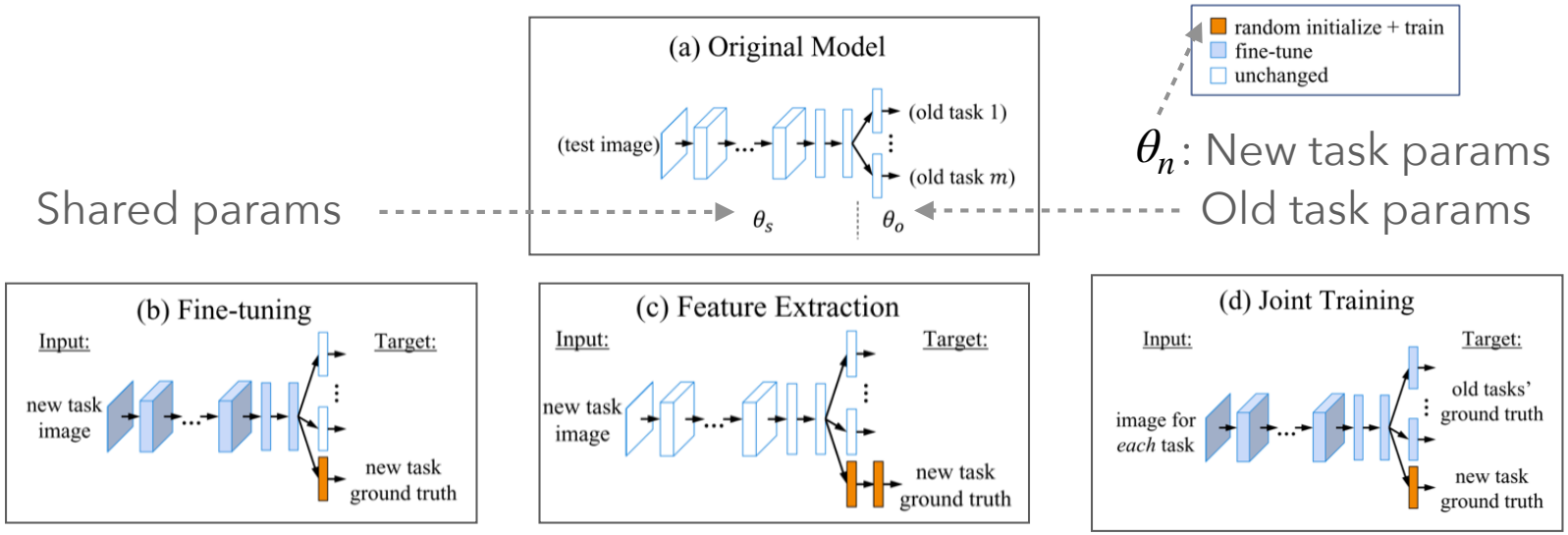
Forgetting previously trained tasks while learning new tasks sequentially



- Main approaches
  - Regularization based
  - Generative replay

Summary of CL Approaches

[Li and Hoeim, ICML 2016; Chen and Liu, 2018]

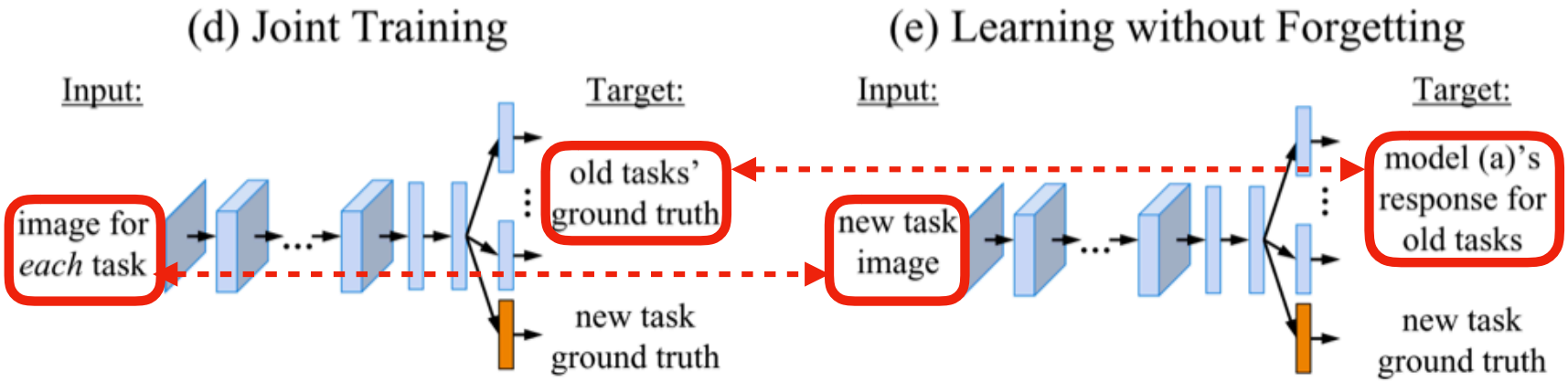


Category	Feature Extraction	Fine-Tuning	Joint Training
New task performance	Medium	Good	Good
Old task performance	Good	Bad	Good
Training efficiency	Fast	Fast	Slow
Testing efficiency	Fast	Fast	Fast
Storage requirement	Medium	Medium	Large
Require previous task data	No	No	Yes

Learning without Forgetting (LwF)

[Li and Hoeim, ICML 2016]

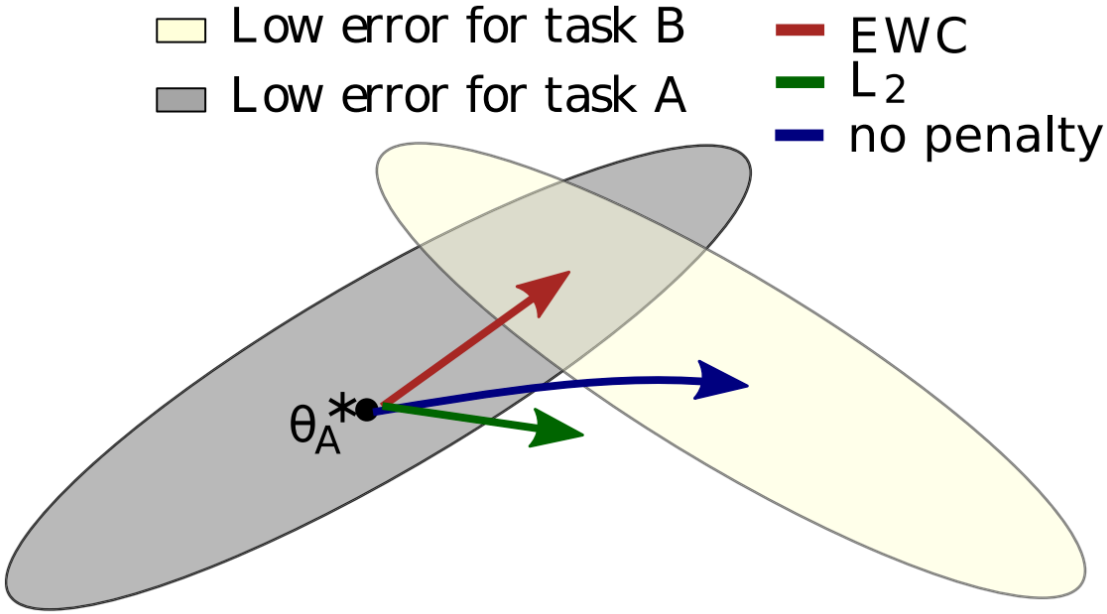
LwF: Training data from old tasks is not available



- Update shared and old task params so that old task output on new task data are preserved
- Constraint on output, rather than on parameters directly
- Experiments on image classification datasets: ImageNet => Scenes

Elastic Weight Consolidation (EWC)

[Kirkpatrick et al, PNAS 2017]



Idea: Don't let *important* parameters change drastically (reduce plasticity)

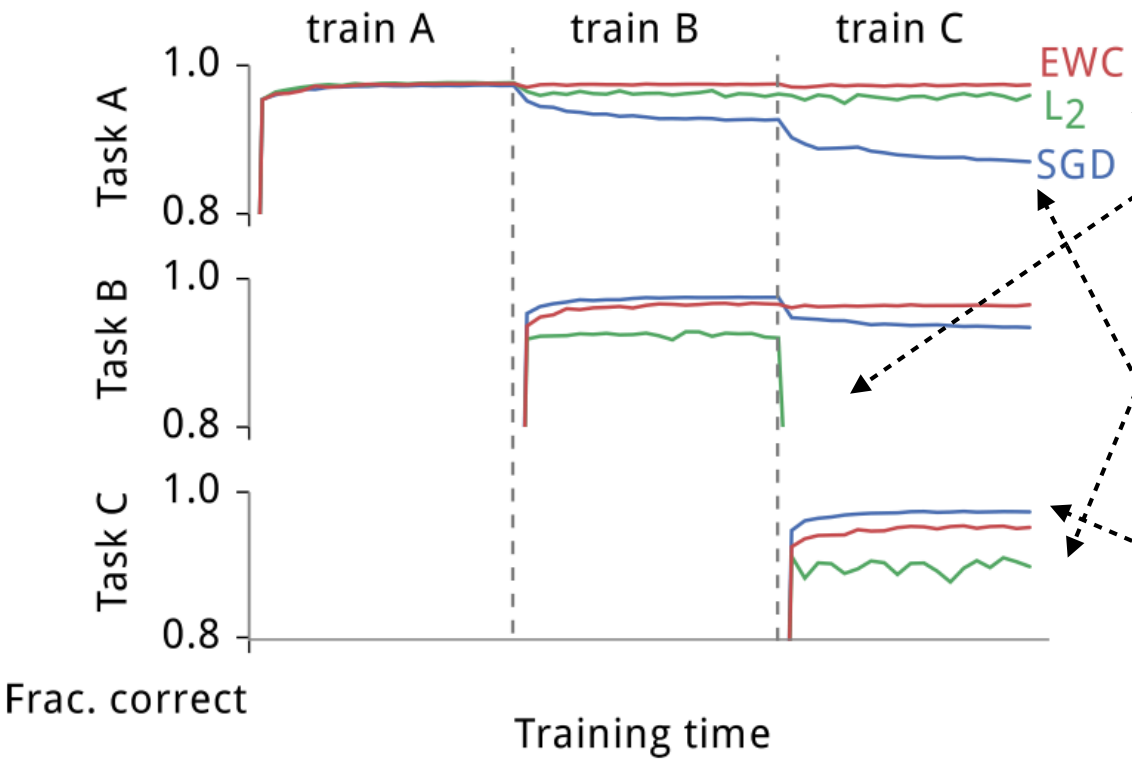
$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{A,i})^2,$$

Task B Loss

- Inspired by research on synaptic consolidation

Elastic Weight Consolidation (EWC)

[Kirkpatrick et al., PNAS 2017]



L2 is too rigid, doesn't allow learning on new tasks => parameter weighting is important

Catastrophic Forgetting in SGD

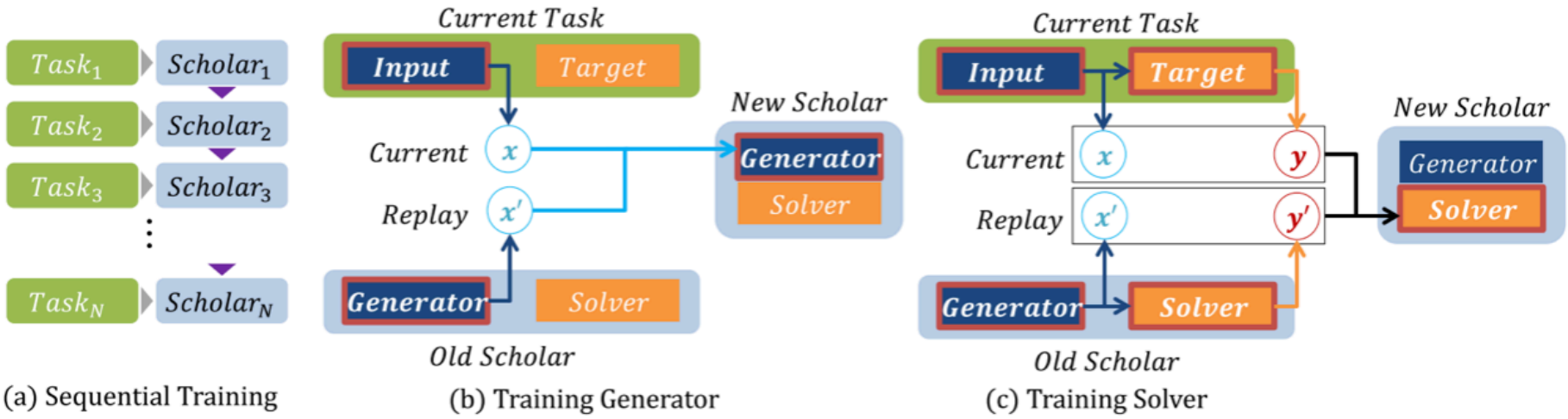
MNIST experiments. New tasks are random pixel permutations.

Deep Generative Replay

[Shin et al., NeurIPS 2017]

Generate old task pseudo data using generative model (e.g., GAN). No exact replay of old task data.

$$L_{train}(\theta_i) = r\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_i}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})] + (1-r)\,\mathbb{E}_{\boldsymbol{x}'\sim G_{i-1}}[L(S(\boldsymbol{x}';\theta_i),S(\boldsymbol{x}';\theta_{i-1}))]$$



(a) Sequential Training (b) Training Generator (c) Training Solver

CL Evaluations

[Kemker et al., AAAI 2018]

	MNIST	CUB-200	AudioSet
Classification Task	Gray Image	RGB Image	Audio
Classes	10	200	100
Feature Shape	784	2,048	1,280
Train Samples	50,000	5,994	28,779
Test Samples	10,000	5,794	5,523
Train Samples/Class	5,421-6,742	29-30	250-300
Test Samples/Class	892-1,135	11-30	43-62

Model	Data Permutation	Incremental Class	Multi-Modal
MLP	0.594	0.085	0.600
EWC	0.586	0.087	**0.913**
PathNet	**0.706**	N/A	0.666
GeppNet	0.284	**0.818**	0.275
GeppNet+STM	0.229	0.790	0.222
FEL	0.234	0.347	0.453

- Three settings
  - Data permutation
  - Incremental Class
  - Multimodal

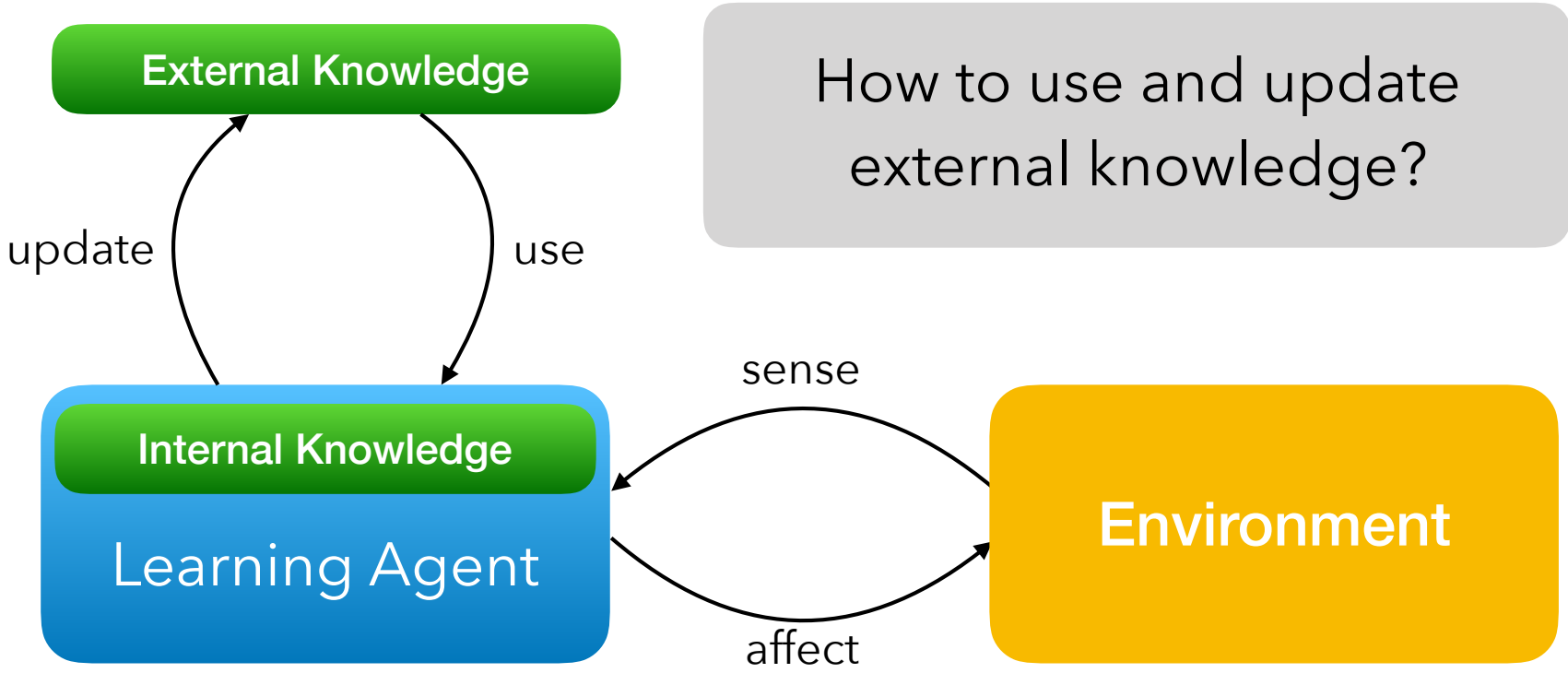
Model	Incremental Class	Similar Data	Dissimilar Data	Memory Efficient	Trains Quickly
MLP	✗	✗	✗	✓	✓
EWC	✗	✗	✓	✓	✓
PathNet	✗	✓	✗	✗	✗
GeppNet	✓	✗	✗	✗	✗
GeppNet+STM	✓	✗	✗	✗	✗
FEL	✗	✗	✗	✗	✓

No single winner. CF is far from being solved.

Research Issues

- Continual Learning and Catastrophic Forgetting
- (External) Knowledge and Reasoning
- Representation Learning
- Self Reflection
- Curriculum Learning

Internal vs External Knowledge



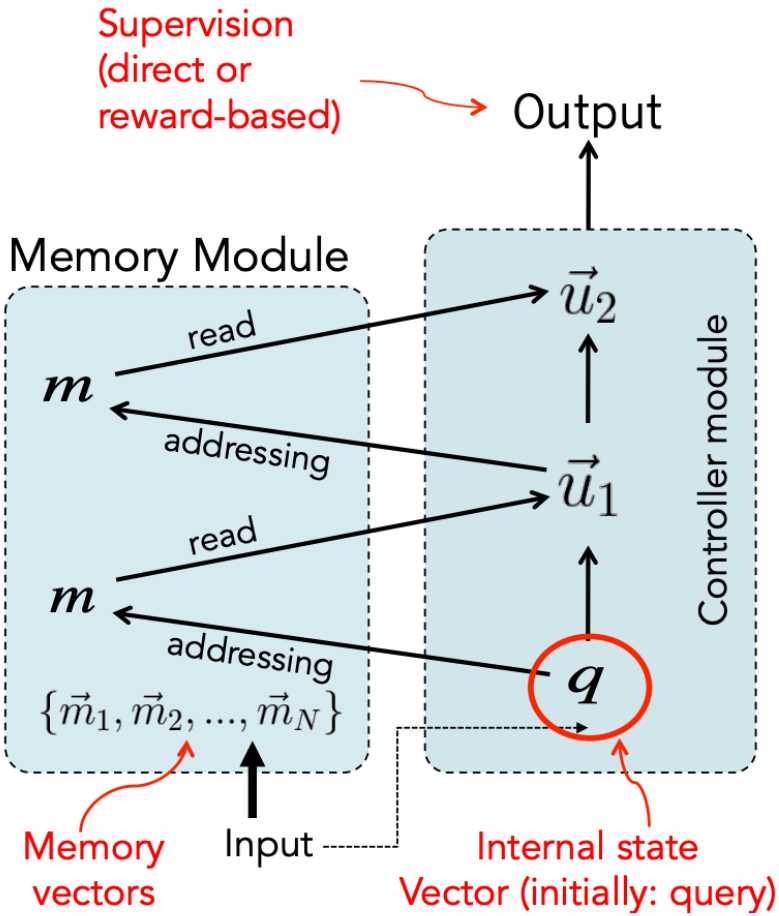
- Two types of external knowledge:
  - memory listing (Memory Networks)
  - relational (Knowledge Graphs)

Memory Networks

[Weston et al., ICLR 2015]

Joe went to the kitchen.
Fred went to the kitchen.
Joe picked up the milk.
Joe travelled to the office.
Joe left the milk there.
Joe went to the bathroom.
Where is the milk now? (A: office)
Where is Joe? (A: bathroom)

- Memory Nets
  - learning with read/write memory
  - Reasoning with Attention and Memory (RAM)

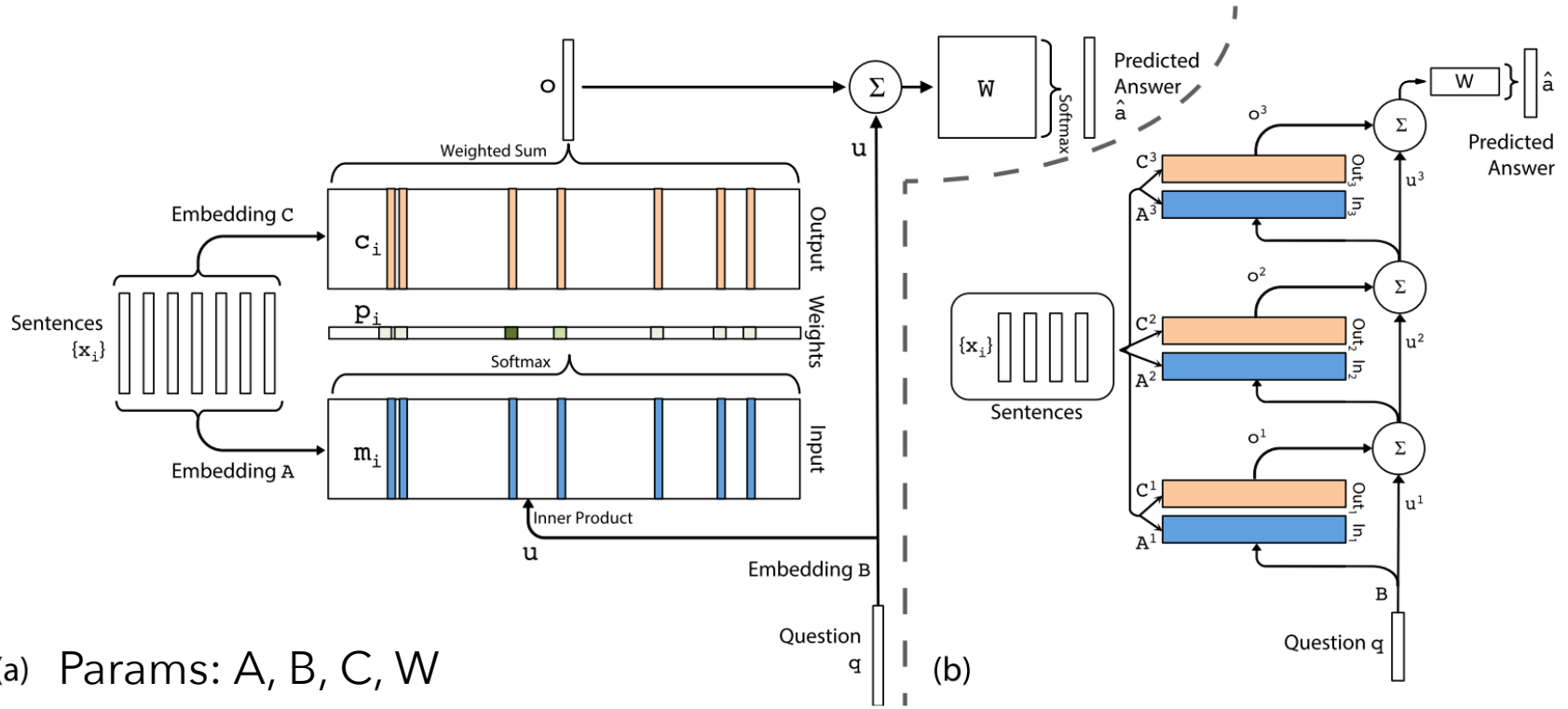


[Figure by Saina Sukhbaatar]

http://www.thespermwhale.com/jaseweston/icml2016/

End2End Memory Networks

[Sukhbaatar et al., NeurIPS 2015]



(a) Params: A, B, C, W

Single Layer

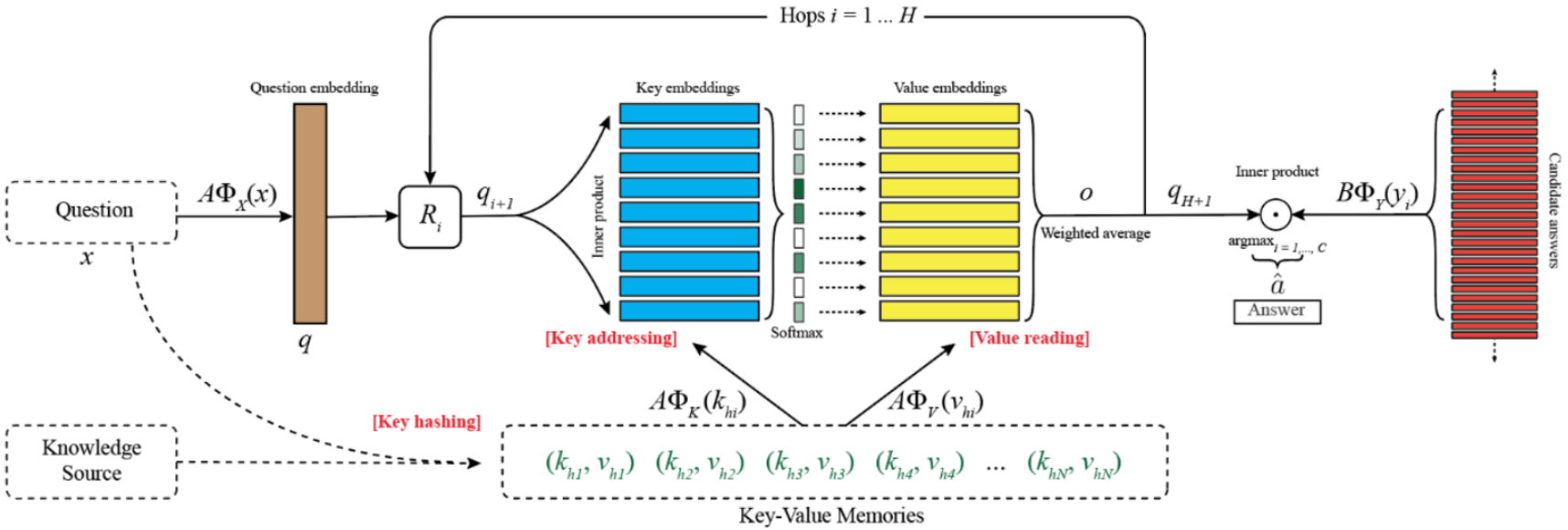
(b)

Three Layers

- Continuous version of the original memory network: soft attention instead of hard
- Supervision only at input-output level, more practical

Key-Value Memory Networks

[Miller et al., EMNLP 2016]

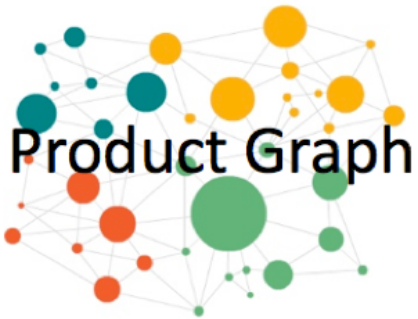


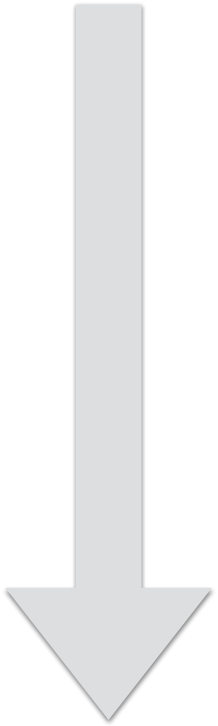
- Structural memory: (key, value), otherwise similar to MemN2N
- Addressing is based on key, reading is based on value

Knowledge Graph Construction Efforts

High Supervision

Amazon

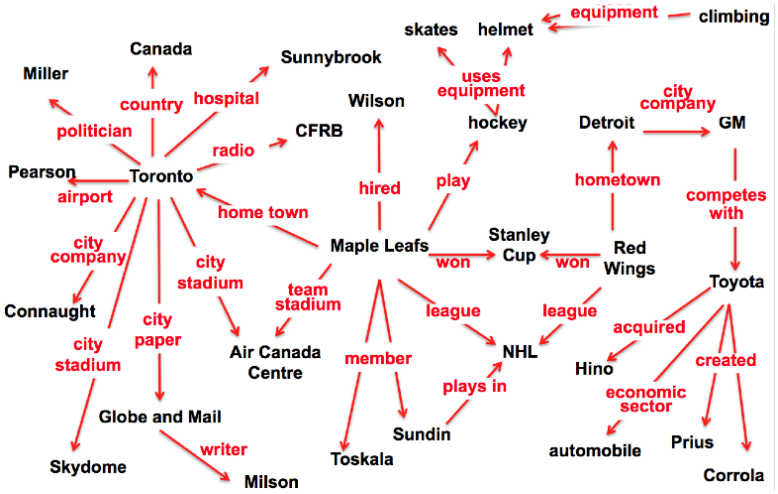


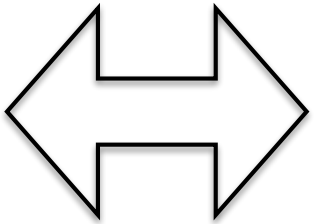


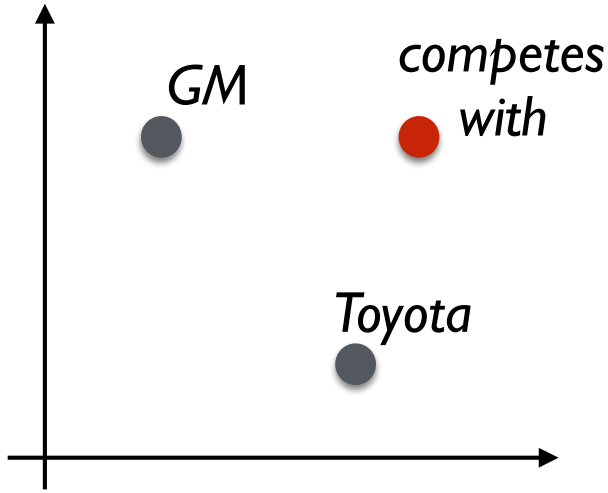
NELL

Low Supervision

Two Views of Knowledge







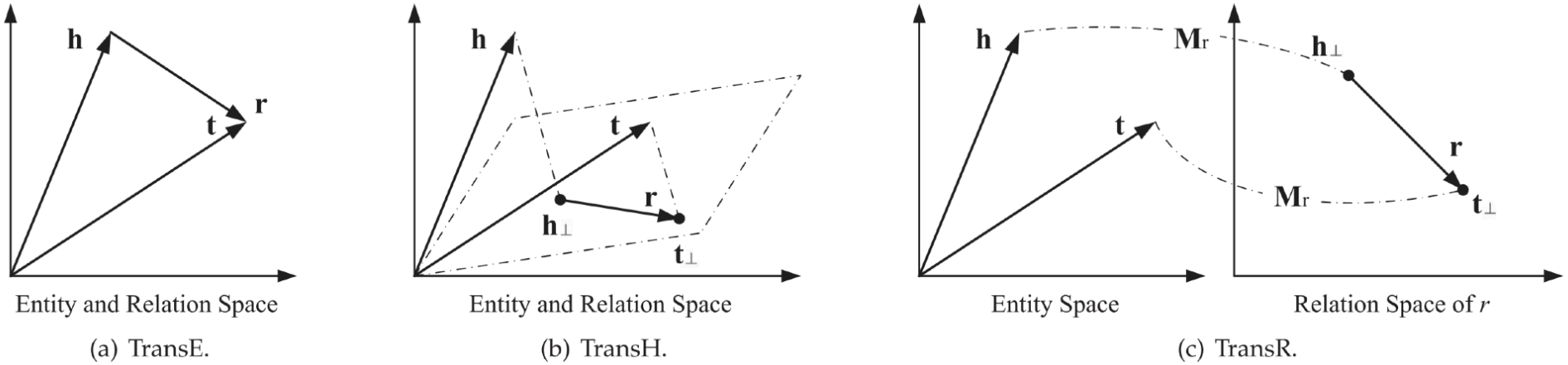
Knowledge Graph

Dense Representations

Knowledge Graph Embedding

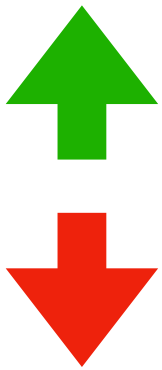
[Surveys: Wang et al., TKDE 2017, ThuNLP]

(h, r, t) = (Barack Obama, presidentOf, USA)



$$h + r \approx t$$

Triple scoring function: $f_r(h, t)$



Positive triples

Negative triples

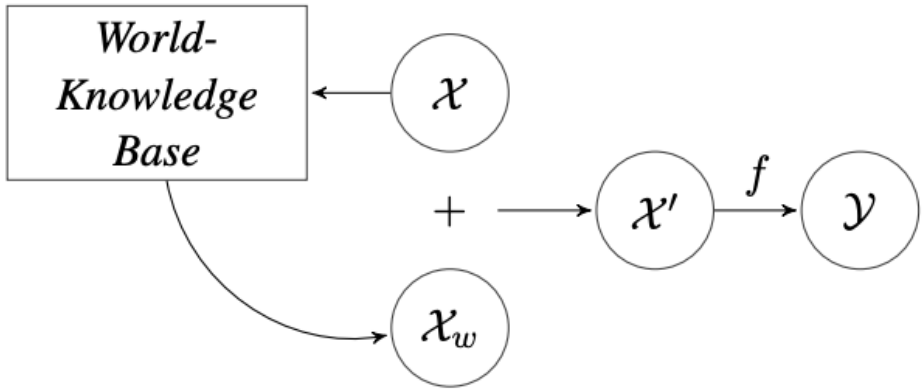
Knowledge Graph Embedding

[Surveys: Wang et al., TKDE 2017, ThuNLP]

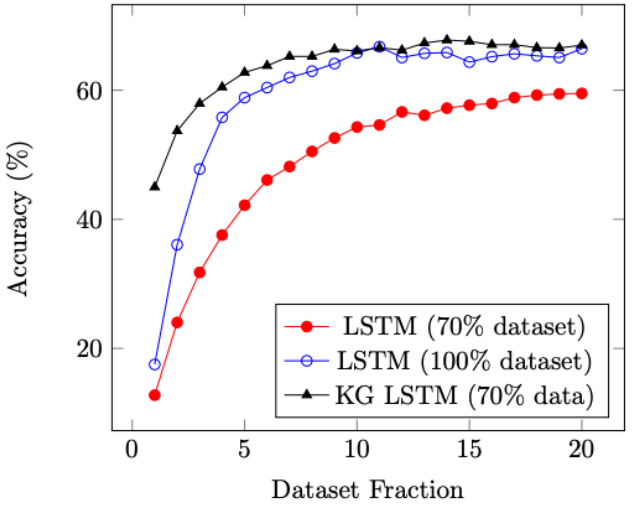
Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h,t)$	Constraints/Regularization
TransE [14]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
TransH [15]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^d$	$-\|(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\|_2^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1$
TransR [16]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r\in\mathbb{R}^{k\times d}$	$-\|\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\|_2^2$	$\|\mathbf{w}_r^\top\mathbf{r}\|/\|\mathbf{r}\|_2\le\epsilon, \|\mathbf{w}_r\|_2=1$ $\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
TransD [50]	$\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$ $\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^k$	$-\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2^2$	$\|\mathbf{M}_r\mathbf{h}\|_2\le 1, \|\mathbf{M}_r\mathbf{t}\|_2\le 1$ $\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\|_2\le 1$ $\|(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2\le 1$
TranSparse [51]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r(\theta_r)\in\mathbb{R}^{k\times d}$ $\mathbf{M}_r^1(\theta_r^1), \mathbf{M}_r^2(\theta_r^2)\in\mathbb{R}^{k\times d}$	$-\|\mathbf{M}_r(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M}_r(\theta_r)\mathbf{t}\|_{1/2}^2$ $-\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\|_{1/2}^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|\mathbf{M}_r(\theta_r)\mathbf{h}\|_2\le 1, \|\mathbf{M}_r(\theta_r)\mathbf{t}\|_2\le 1$ $\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}\|_2\le 1, \|\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\|_2\le 1$
TransM [52]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\theta_r\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
ManifoldE [53]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2-\theta_r^2)^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
TransF [54]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$(\mathbf{h}+\mathbf{r})^\top\mathbf{t}+(\mathbf{t}-\mathbf{r})^\top\mathbf{h}$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
TransA [55]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d, \mathbf{M}_r\in\mathbb{R}^{d\times d}$	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)^\top\mathbf{M}_r(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
KG2E [45]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\Sigma_h)$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\Sigma_t)$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$ $\Sigma_h,\Sigma_t\in\mathbb{R}^{d\times d}$	$\mathbf{r}\sim\mathcal{N}(\boldsymbol{\mu}_r,\Sigma_r)$ $\boldsymbol{\mu}_r\in\mathbb{R}^d, \Sigma_r\in\mathbb{R}^{d\times d}$	$-\mathrm{tr}(\Sigma_r^{-1}(\Sigma_h+\Sigma_t))-\boldsymbol{\mu}^\top\Sigma_r^{-1}\boldsymbol{\mu}-\ln\frac{\det(\Sigma_r)}{\det(\Sigma_h+\Sigma_t)}$ $-\boldsymbol{\mu}^\top\Sigma^{-1}\boldsymbol{\mu}-\ln(\det(\Sigma))$ $\boldsymbol{\mu}=\boldsymbol{\mu}_h+\boldsymbol{\mu}_r-\boldsymbol{\mu}_t$ $\Sigma=\Sigma_h+\Sigma_r+\Sigma_t$	$\|\mathbf{M}_r\|_F\le 1, [\mathbf{M}_r]_{ij}=[\mathbf{M}_r]_{ji}\ge 0$ $\|\boldsymbol{\mu}_h\|_2\le 1, \|\boldsymbol{\mu}_t\|_2\le 1, \|\boldsymbol{\mu}_r\|_2\le 1$ $c_{min}\mathbf{I}\le\Sigma_h\le c_{max}\mathbf{I}$ $c_{min}\mathbf{I}\le\Sigma_t\le c_{max}\mathbf{I}$ $c_{min}\mathbf{I}\le\Sigma_r\le c_{max}\mathbf{I}$
TransG [46]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\sigma_h^2\mathbf{I})$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma_t^2\mathbf{I})$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$	$\boldsymbol{\mu}_r^i\sim\mathcal{N}(\boldsymbol{\mu}_t-\boldsymbol{\mu}_h,(\sigma_h^2+\sigma_t^2)\mathbf{I})$ $\mathbf{r}=\sum_i\pi_r^i\boldsymbol{\mu}_r^i\in\mathbb{R}^d$	$\sum_i\pi_r^i\exp\left(-\frac{\|\boldsymbol{\mu}_h+\boldsymbol{\mu}_r^i-\boldsymbol{\mu}_t\|_2^2}{\sigma_h^2+\sigma_t^2}\right)$	$\|\boldsymbol{\mu}_h\|_2\le 1, \|\boldsymbol{\mu}_t\|_2\le 1, \|\boldsymbol{\mu}_r^i\|_2\le 1$
UM [56]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	—	$-\|\mathbf{h}-\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
SE [57]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{d\times d}$	$-\|\mathbf{M}_r^1\mathbf{h}-\mathbf{M}_r^2\mathbf{t}\|_1$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$

Using KG for Document Classification

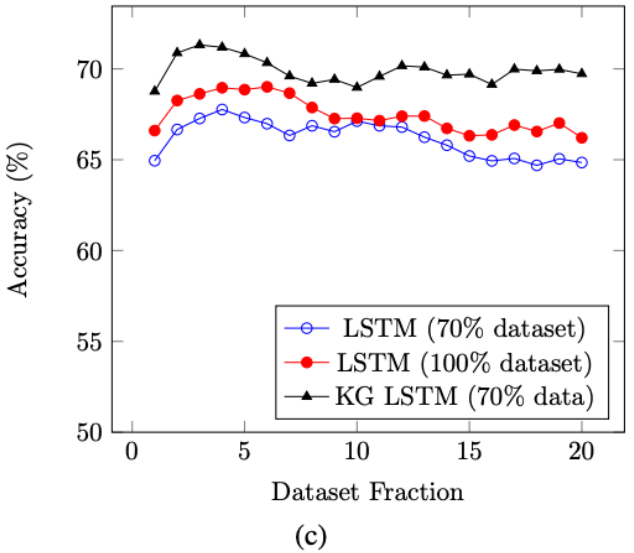
[Annervaz et al., NAACL 2018]



Incorporation of word knowledge helps improve deep learning performance

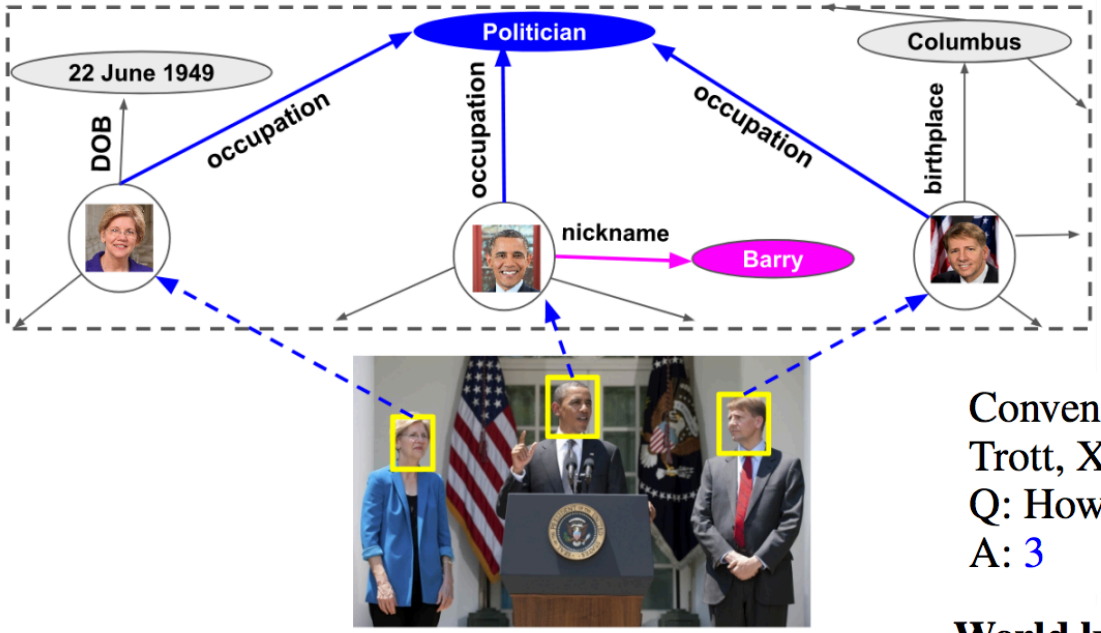


News20



SNLI

Knowledge-aware Visual Question Answering [Shah et al., AAAI 2019]



KVQA

[http://malllabiisc.github.io/resources/kvqa/]

New Dataset for Knowledge-aware Computer Vision

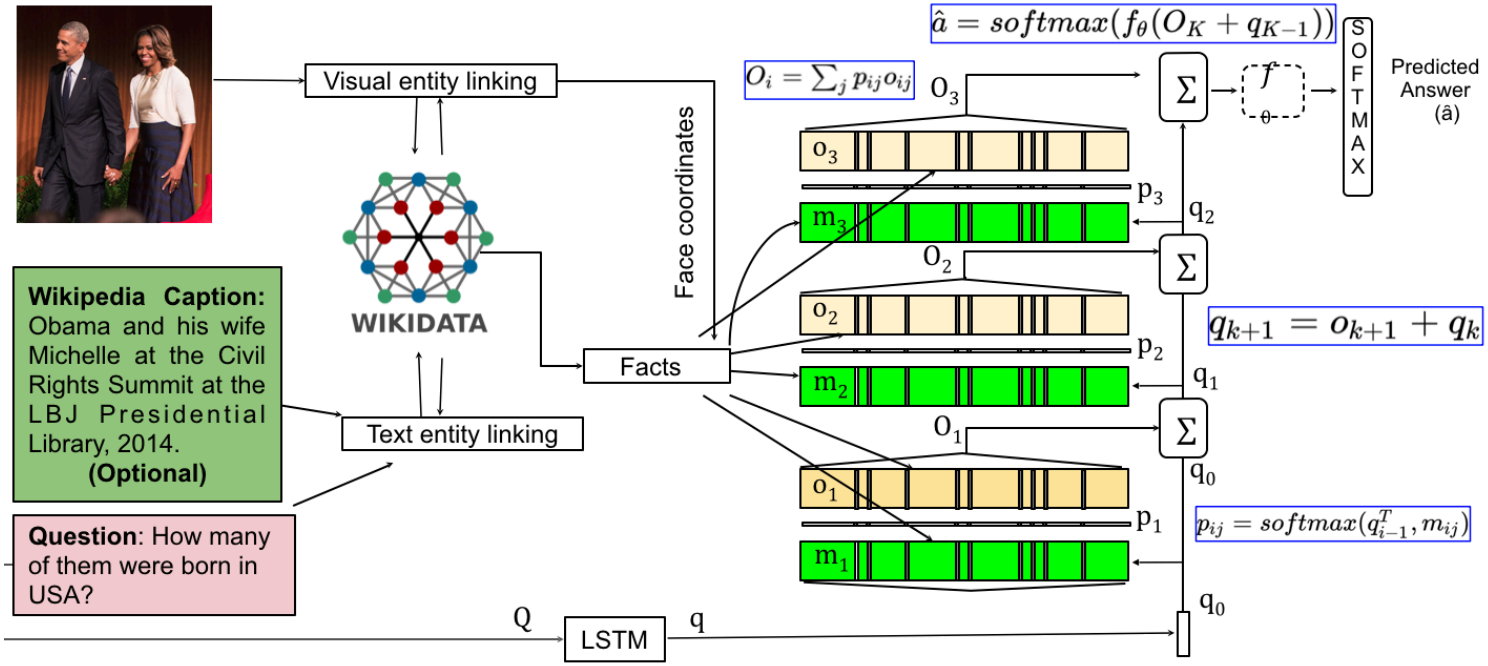
KVQA Dataset

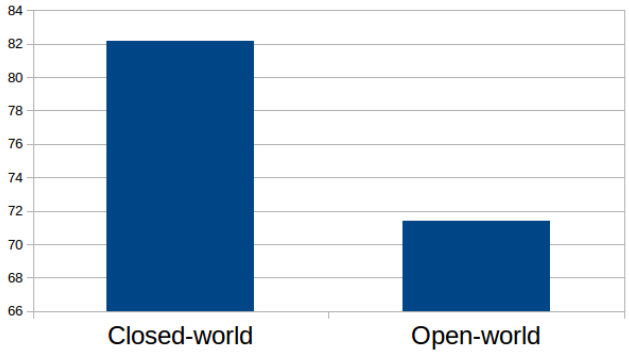
- 24k+ images
- 19.5k+ unique answers
- 183k+ QA pairs

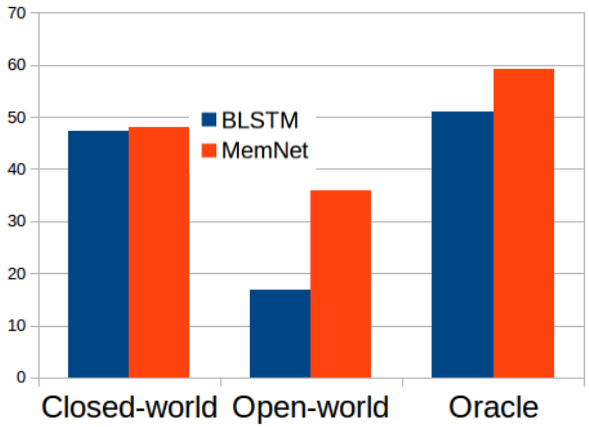
Conventional VQA (Antol et al. 2015; Goyal et al. 2017; Trott, Xiong, and Socher 2018)
Q: How many people are there in the image?
A: 3

World knowledge-enabled VQA (this paper):
Q: Who is to the left of Barack Obama?
A: Richard Cordray
Q: Do people in the image have common occupation?
A: Yes
Q: Who among the people in the image is called by the nickname Barry?
A: Person in the center

Approach







Category	ORG	PRP	Category	ORG	PRP
Spatial	48.1	47.2	Multi-rel.	45.2	44.2
1-hope	61.0	60.2	Subtraction	40.5	38.0
Multi-hop	53.2	52.1	Comparison	50.5	49.0
Boolean	75.1	74.0	Counting	49.5	48.9
Intersect.	72.5	71.8	Multi-entity	43.5	43.0

Table 5: VQA results on different categories of questions in paraphrased (PRP) and original version (ORG) of questions in KVQA tested using MemNet. Please see the "VQA over KG" section for more details.

Requires reasoning over KG. Significant room for improvement.

Research Issues

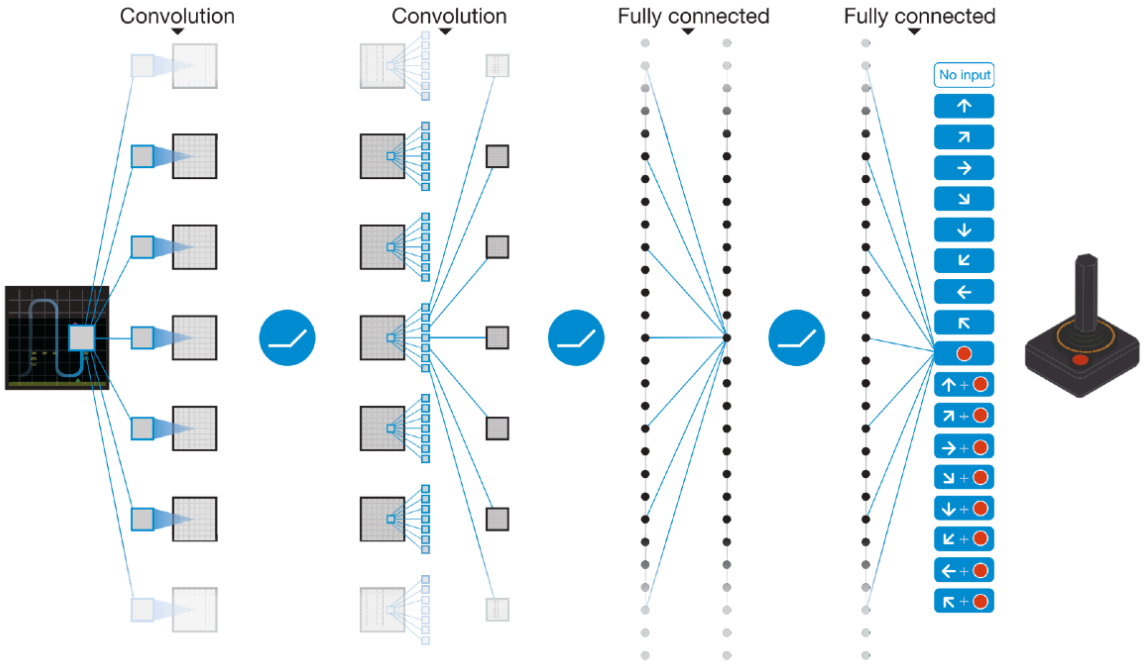
- Continual Learning and Catastrophic Forgetting
- (External) Knowledge and Reasoning
- Representation Learning
  - States
  - Sequences
- Self Reflection
- Curriculum Learning

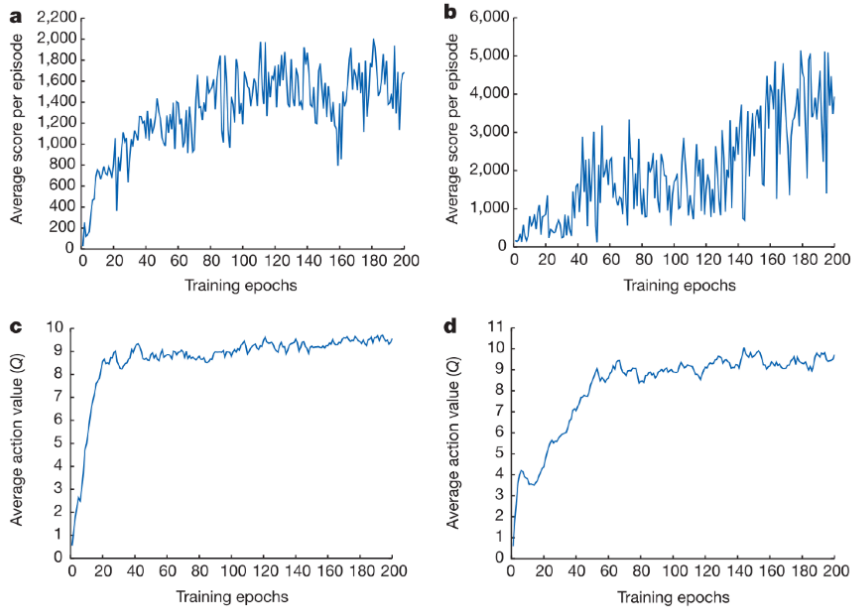
Deep Reinforcement Learning

[Mnih et al., NeurIPS 2013, Mnih et al., Nature 2015]

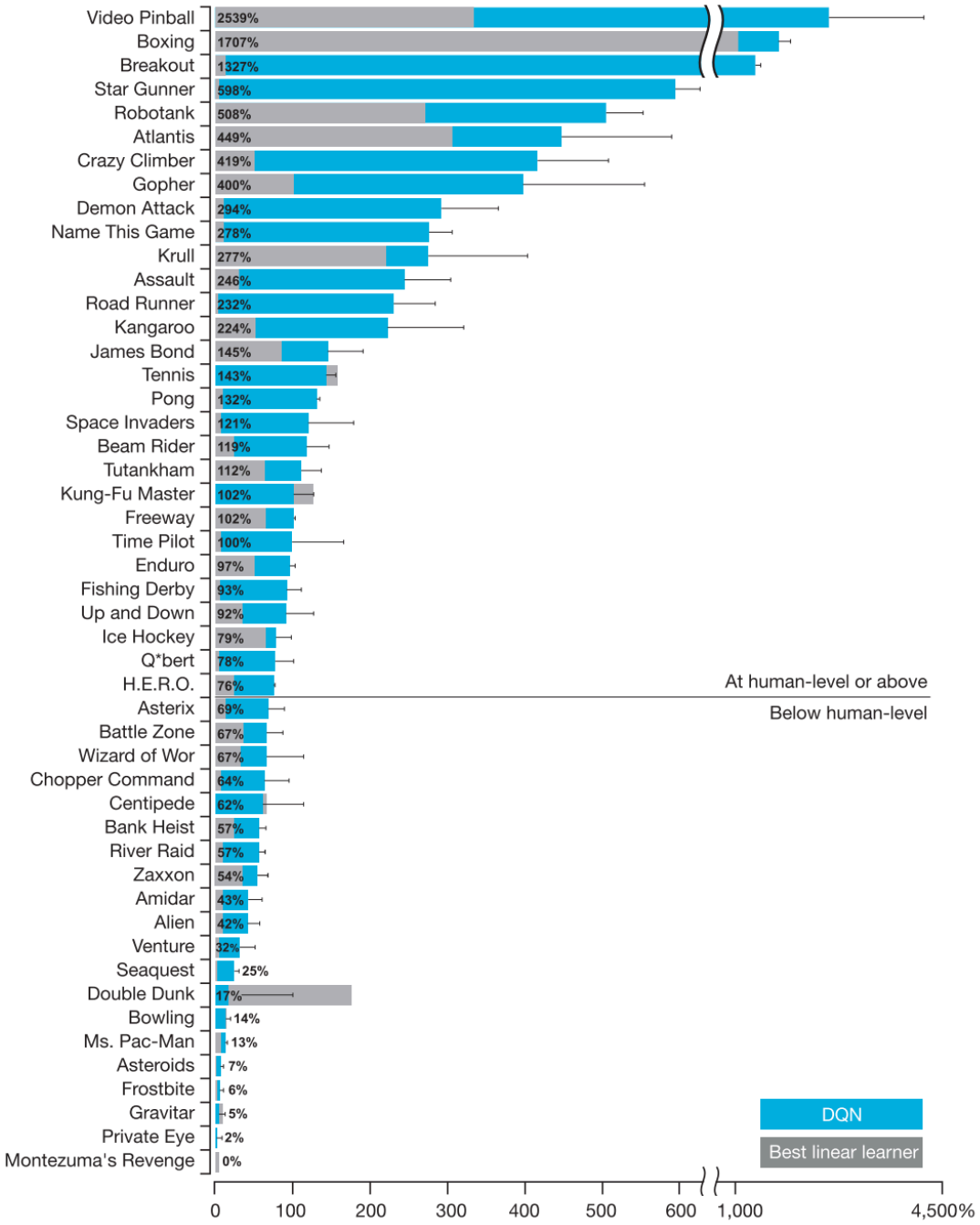
$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \,|\, s_t = s,\, a_t = a,\, \pi\right]$$

Deep Q Network (DQN) $Q(s, a; \theta_i)$



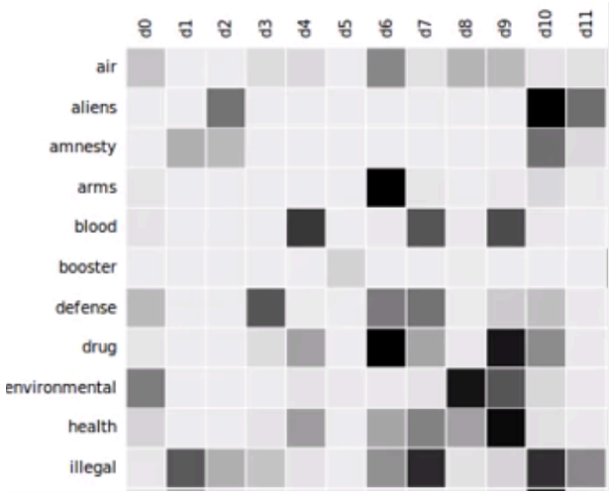


DQN on 49 Atari Games

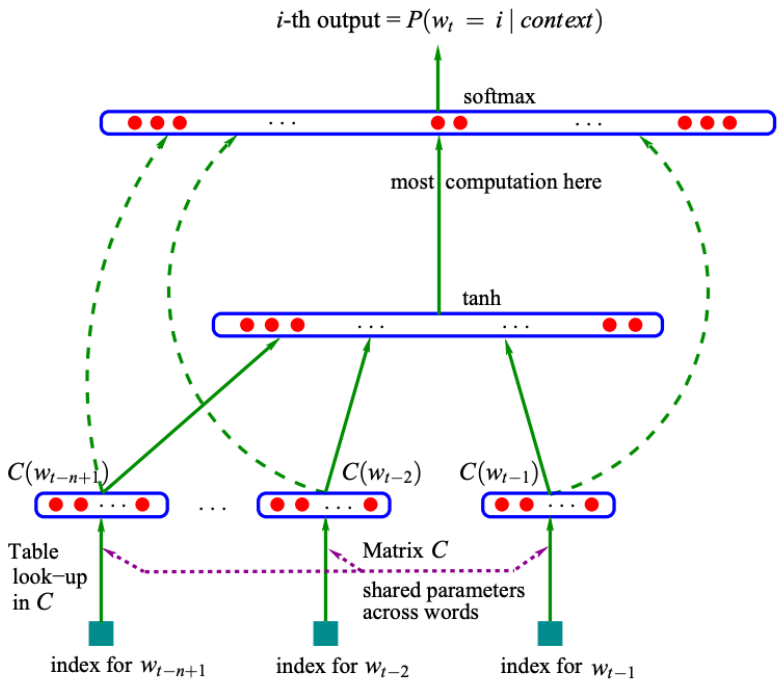


- More predictive state representation using deep CNN
- Trained on random samples of past plays: Experience replay
- Super-human performance on many tasks using same network (trained separately)
- Limitation: requires lots of replays to learn

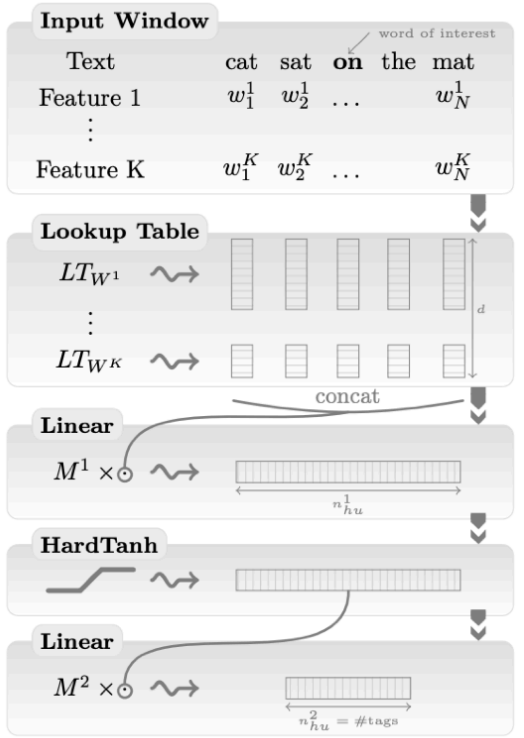
Learning Word Meanings



[Deerwester et al., 1988]



[Bengio et al., 2003]

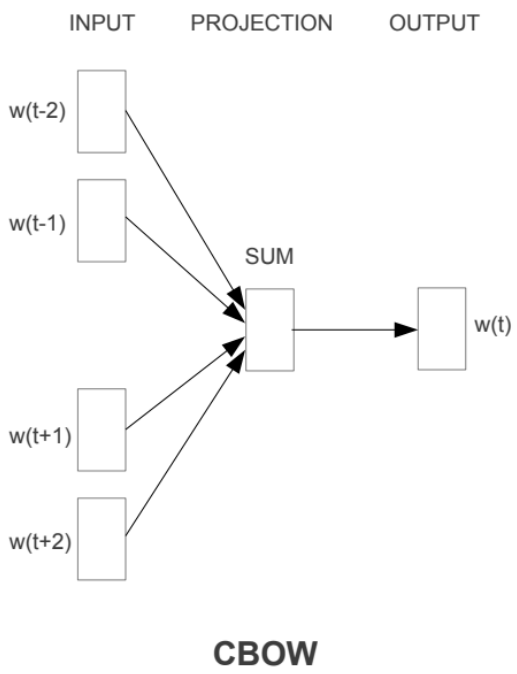


[Collobert et al., 2011]

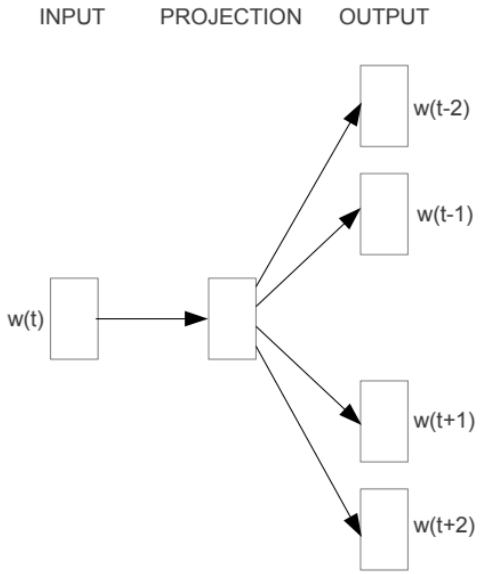
Representing word meanings as vectors utilizing its context has a long history [Harris, 1954]

Representation Learning in NLP

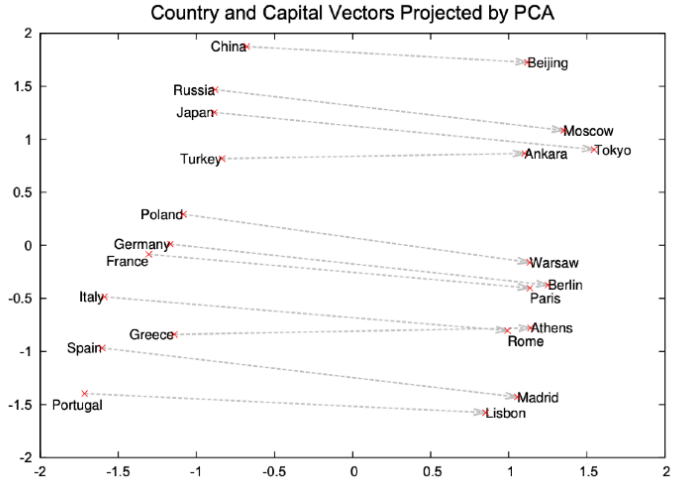
Word2Vec [Mikolov et al., 2013a; Mikolov et al., NeurIPS 2013b]



CBOW



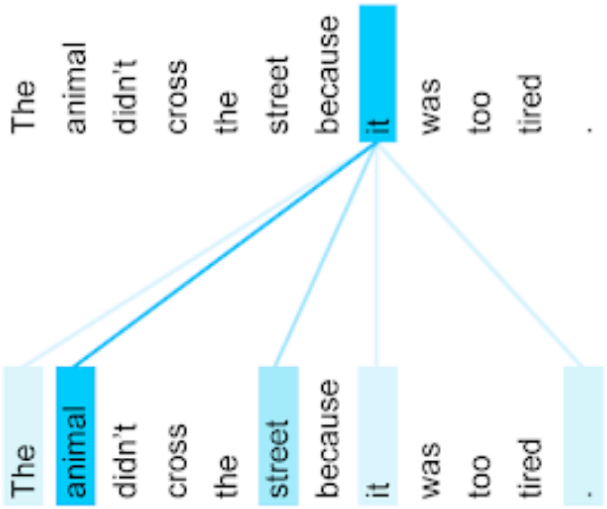
Skip-gram

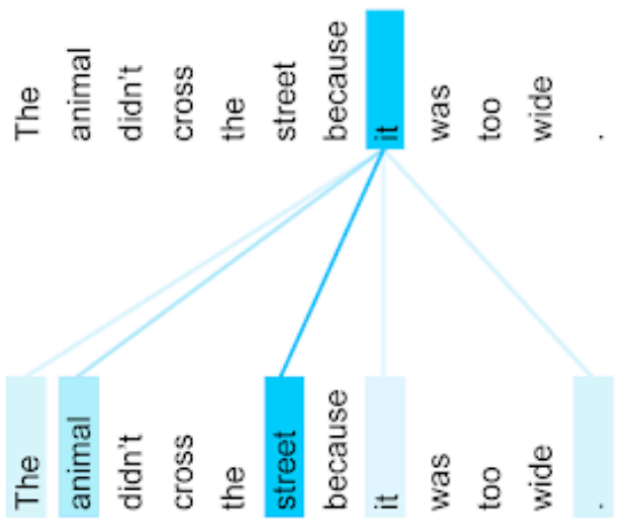


- Learn word embeddings by creating word prediction problems out of unlabeled corpus
- Big impact in NLP, lots of subsequent work, e.g., Glove,

Representations using Self-Attention

Transformers [Vaswani et al., NeurIPS 2018]





Self Attention

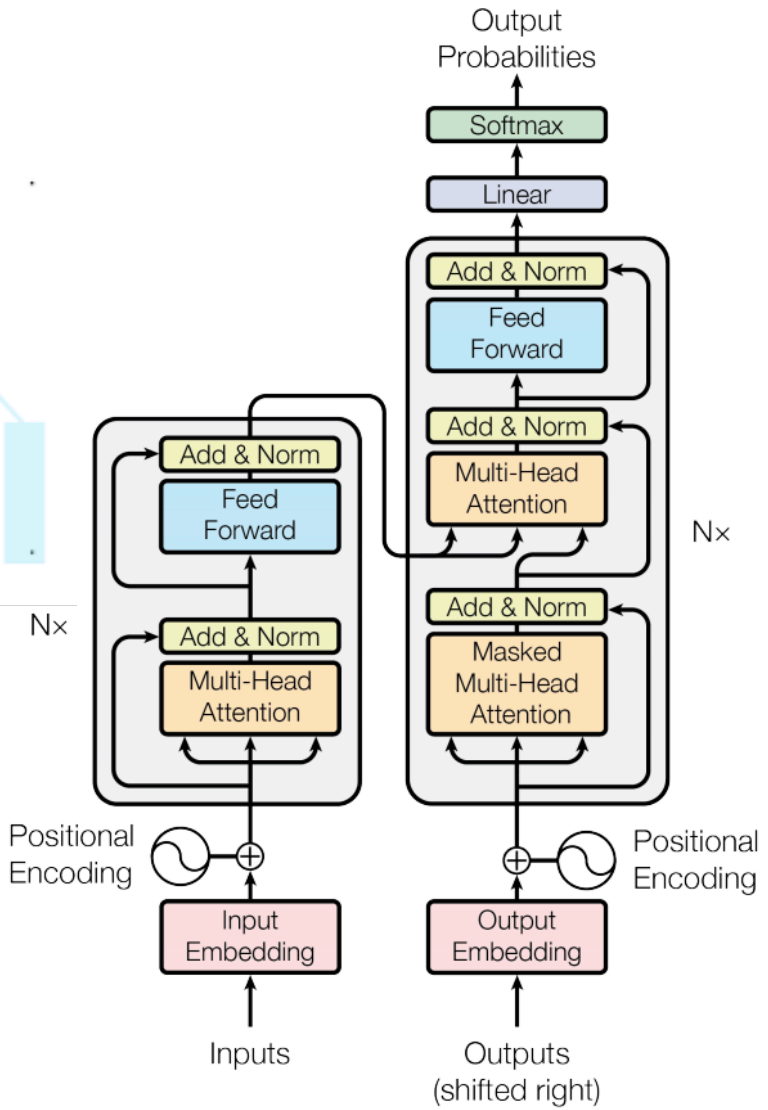
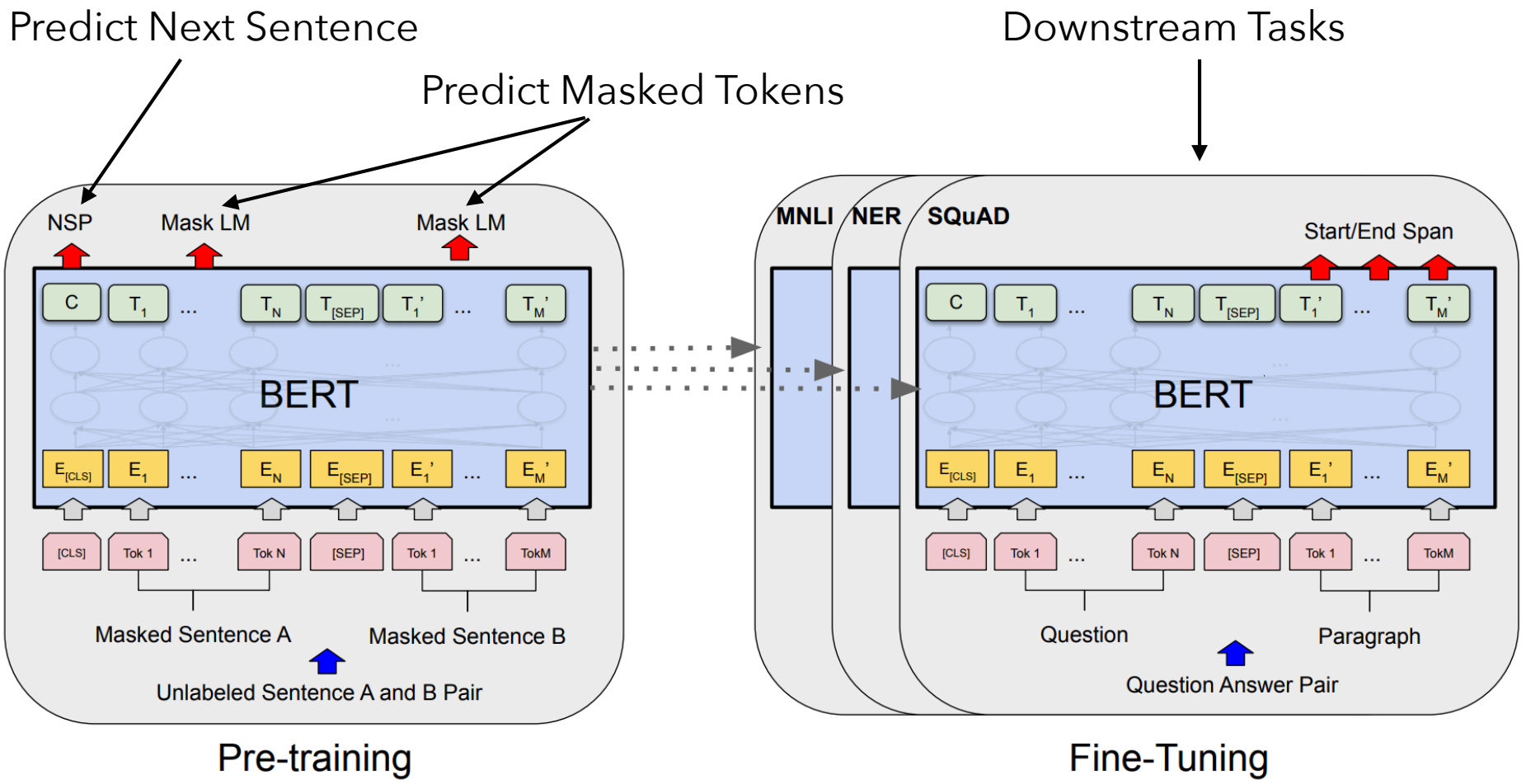
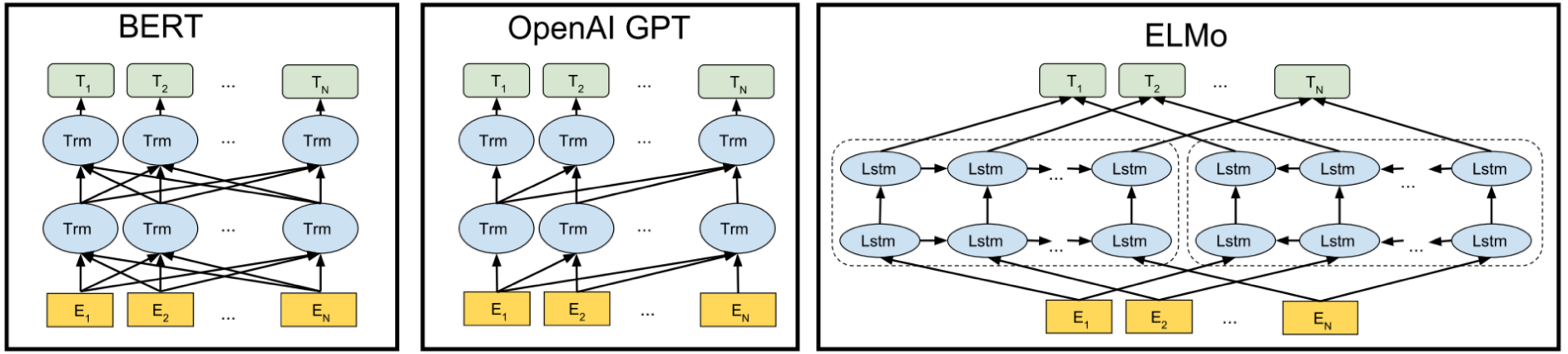


Image Credit: https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Representation Learning in NLP

BERT [Devlin et al., NAACL 2019]





System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

https://gluebenchmark.com/leaderboard/

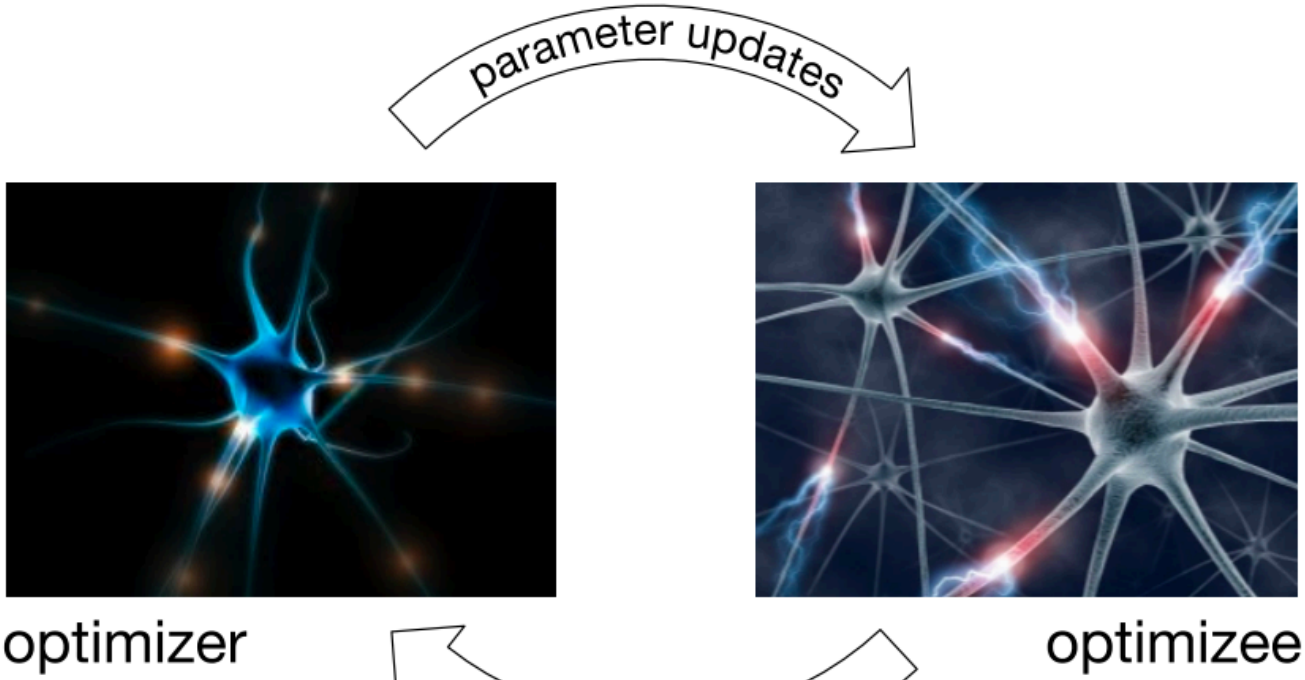
Pre-trained embeddings fine tuned further can be an effective transfer model

Research Issues

- Continual Learning and Catastrophic Forgetting
- (External) Knowledge and Reasoning
- Representation Learning
- Self Reflection
- Curriculum Learning

Learning to Learn by GD by GD

[Andrychowicz et al., NeurIPS 2016]



$$\theta^*(f, \phi)$$

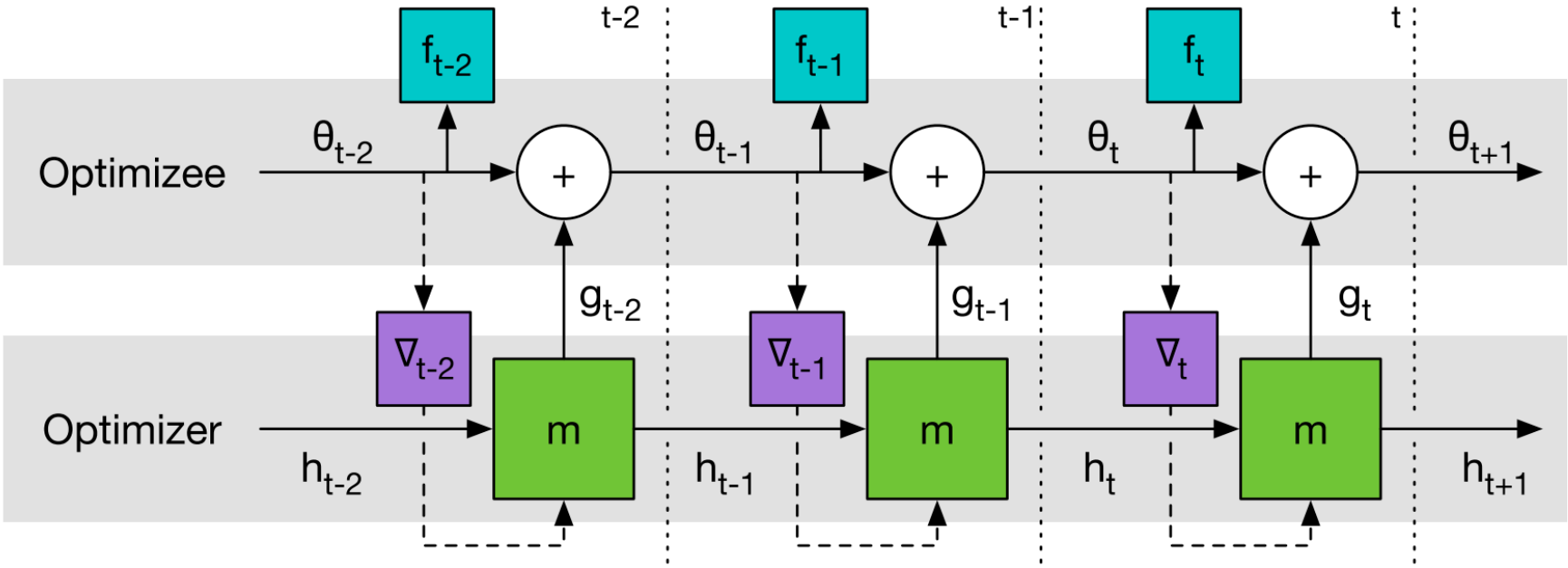
Optimizee parameters

Optimizer parameters

$$\theta_{t+1} = \theta_t + g_t(\nabla f(\theta_t), \phi)$$

Learning to Learn by GD by GD

[Andrychowicz et al., NeurIPS 2016]



$$\mathcal{L}(\phi) = \mathbb{E}_f \left[\sum_{t=1}^{T} w_t f(\theta_t) \right] \qquad \text{where} \qquad \theta_{t+1} = \theta_t + g_t \,,$$

RNN

$$\begin{bmatrix} g_t \\ h_{t+1} \end{bmatrix} = m(\nabla_t, h_t, \phi) \,.$$

Learning Plateaus

- Learning Plateau: a point where further learning iteration doesn't help
- How to detect learning plateaus?
  - detect learning impasse (e.g., SOAR)
  - check change in learning parameters or other metric (e.g., consistency [Platanios et al., 2014])
- How to resolve learning plateaus?
  - switch from exploitation to exploration (especially if local optimum)
  - induce new learning task to resolve impasse (as in SOAR)
  - update knowledge representation
  - ask for help (humans or other agents)

Research Issues

- Continual Learning and Catastrophic Forgetting
- (External) Knowledge and Reasoning
- Representation Learning
- Self Reflection
- Curriculum Learning

Curriculum Learning

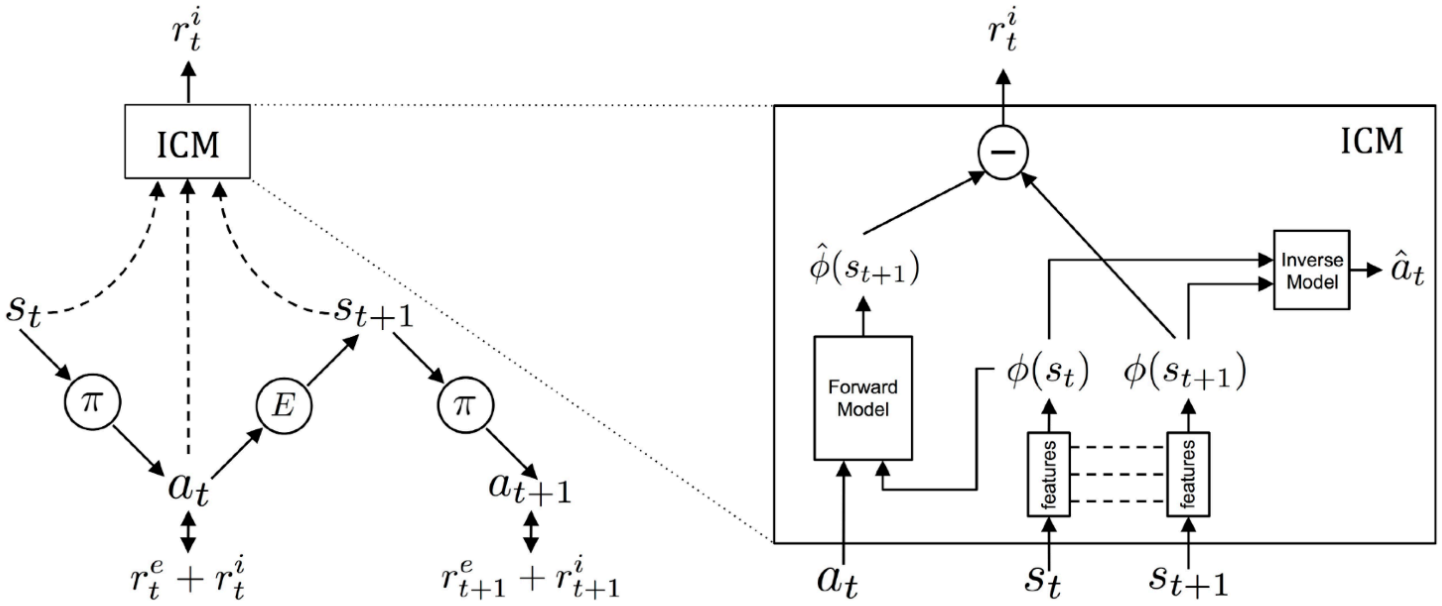
[Bengio et al., ICML 2009]

Start small (or easy), then gradually increase difficulty

- Previously explored in cognitive science [Elman 1993], animal training "*shaping*" [Skinner, 1958]
- Can help with speed and quality of optimization (especially in non-convex settings)
- Curriculum Learning in NELL: relation induction, Horn clause learning, etc.
- Challenges: defining what is easy, determining curriculum order => addressed in [Graves et al., ICML 2017]

Curiosity-driven Learning

[Pathak et al., ICML 2017; Burda et al., ICLR 2019]



- Curiosity is modeled as the model's ability to predict consequences of own action
- Useful with very sparse or no external reward
- However, requires repeated interactions with the environment

Research Issues

- Continual Learning and Catastrophic Forgetting
- (External) Knowledge and Reasoning
- Representation Learning
- Self Reflection
- Curriculum Learning

Resources

- Books & websites
  - Lifelong Machine Learning [Chen and Liu, 2018]
  - Learning to Learn [Thrun 1998]
  - LifeLongML.org
  - The SOAR Cognitive Architecture [Laird, 2012]
- Surveys
  - Continual learning in Neural Networks [Parisi et al., 2019]
  - Lifelong Learning [Silver, 2013]
  - KG Embedding [Wang et al., 2017]

Resources

- Recent Workshops & Tutorials
  - ICML 2018 Workshop on Lifelong RL
  - NeurIPS 2018 MetaLearn
  - NeurIPS 2018 Workshop on Continual Learning
  - NeurIPS 2018 Tutorial on AutoML
  - ICML 2019 Workshop on MTL and Lifelong RL
  - ICML 2019 Workshop on Adaptive and MTL

PhD Thesis Topics in NEL

- What is the effect of different types of coupling constraints (e.g., output coupling, parameter coupling, coupling across time) on learning?
- How to perform coupled learning at scale?
- How should a NEL agent add additional learning tasks?
- Given unlabeled data, is estimating accuracy inherently harder than learning?
- How to incorporate curiosity in a NEL agent?
- How to build a cooperative community of NEL agents?
- What are the sufficient modes of self-reflection?
- How can a NEL agent exploit multiple modalities?
- How should a NEL agent communicate with humans?

Thanks!

https://sites.google.com/site/neltutorialicml19/

tom.mitchell@cs.cmu.edu, ppt@iisc.ac.in