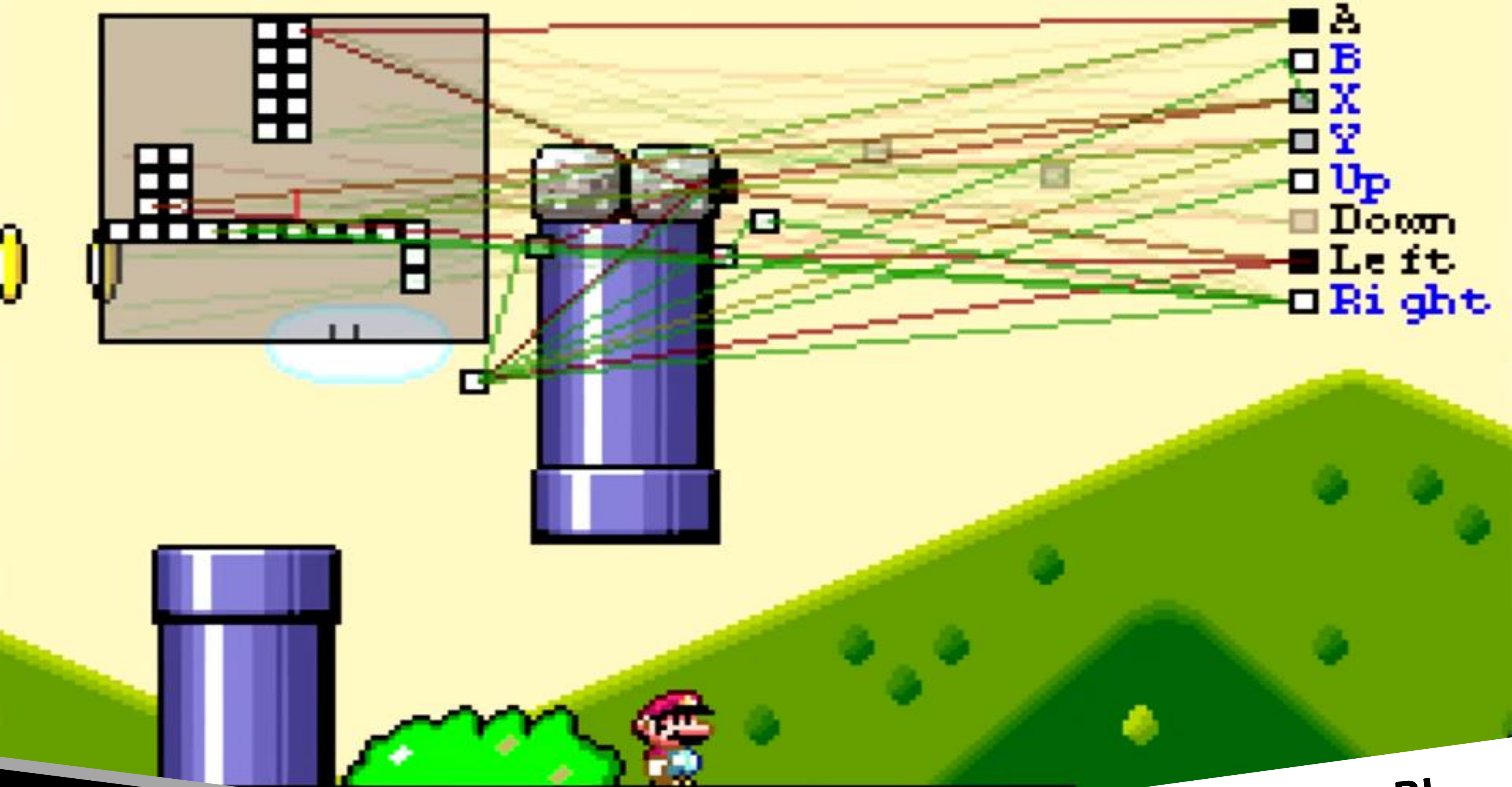


More about RL

Apr 23th, 2019

Applied Deep Learning

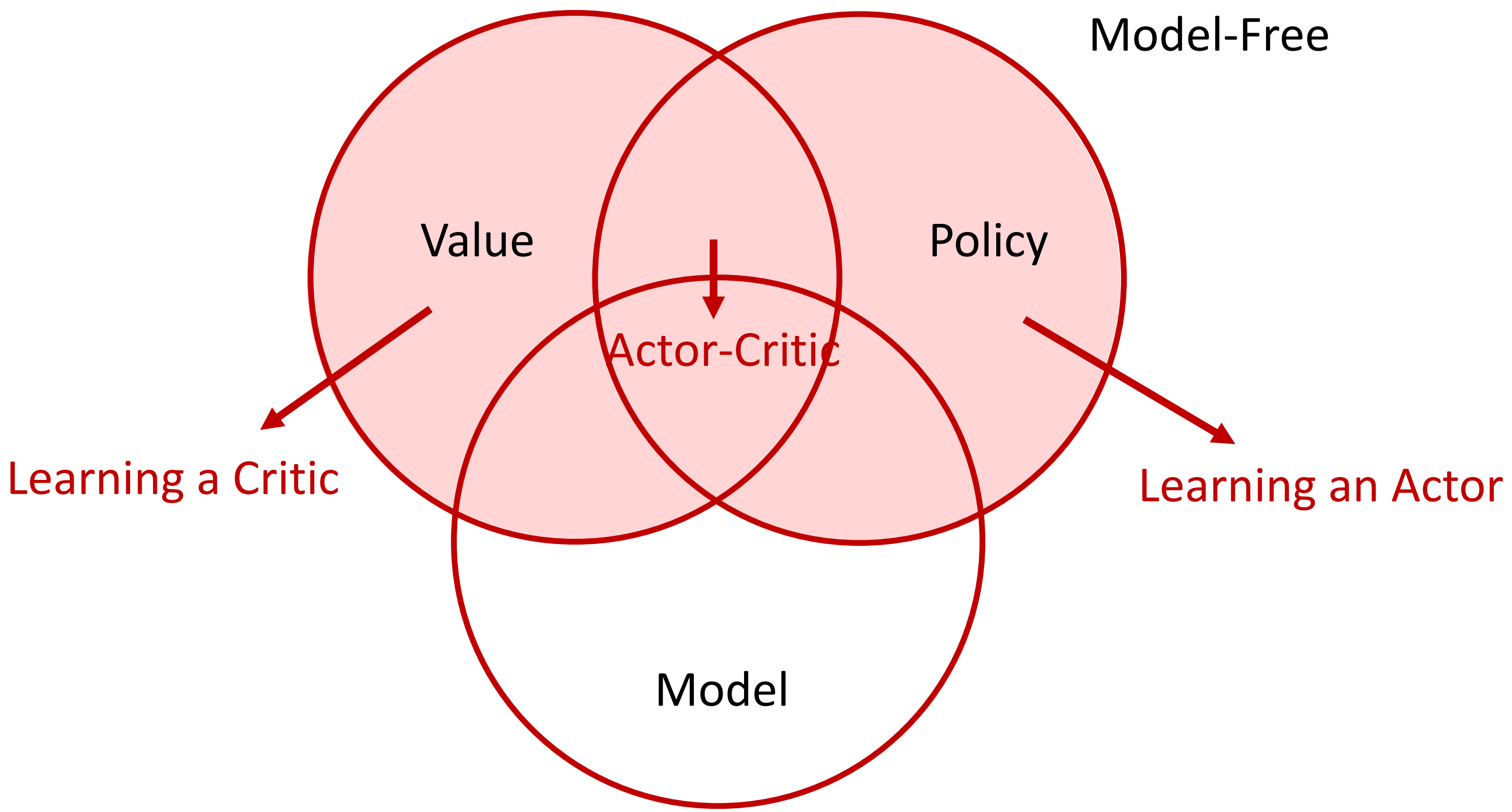
YUN-NUNG (VIVIAN) CHEN

HTTP://ADL.MIULAB.TW

國立臺灣大學
National Taiwan University

Slides credited from Dr. Hung-Yi Lee

RL Agent Taxonomy



Model-Based

Agent’s Representation of the Environment

Model

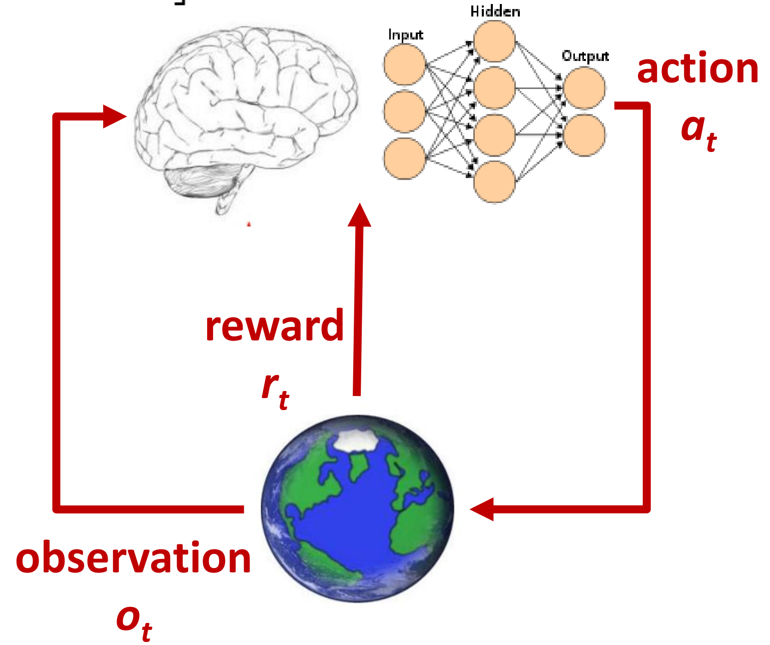
A model predicts what the environment will do next

- *P* predicts the next state

$$P^a_{ss'} = \mathbb{P}[s_{t+1} = s' \mid s_t = s, a_t = a]$$

- *R* predicts the next immediate reward

$$R^a_s = \mathbb{E}[r_{t+1} \mid s_t = s, a_t = a]$$

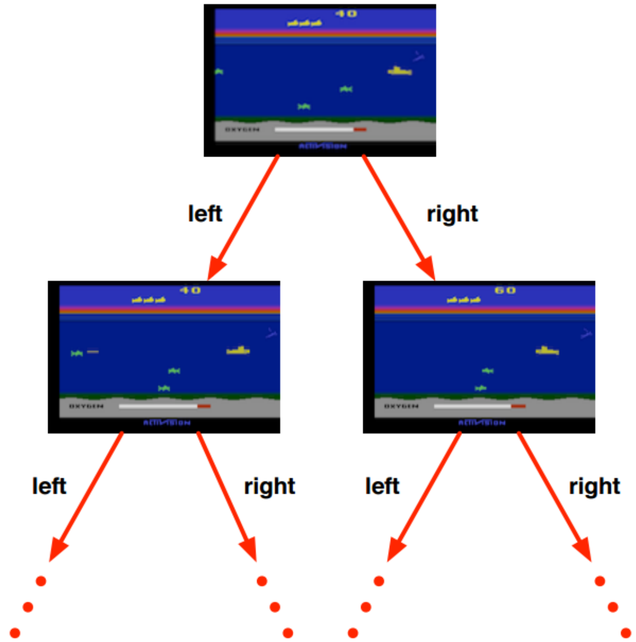


Model-Based Deep RL

Goal: learn a transition model of the environment and plan based on the transition model

$$p(r, s' \mid s, a)$$

Objective is to maximize the measured goodness of model



Model-based deep RL is challenging, and so far has failed in Atari

Issues for Model-Based Deep RL

Compounding errors

- Errors in the transition model compound over the trajectory
- A long trajectory may result in totally wrong rewards

Deep networks of value/policy can “plan” implicitly

- Each layer of network performs arbitrary computational step
- n-layer network can “lookahead” n steps

Model-Based Deep RL in Go

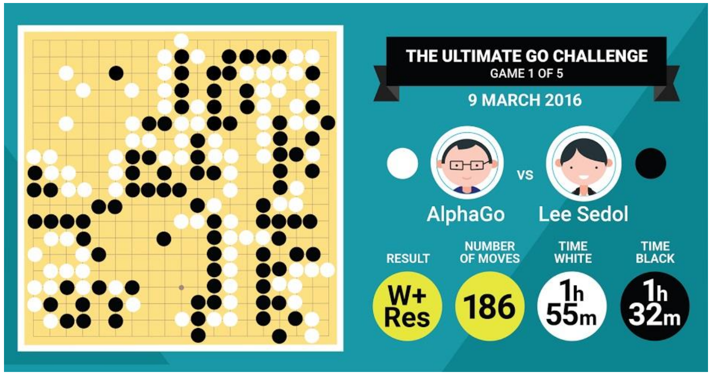
Monte-Carlo tree search (MCTS)

- MCTS simulates future trajectories
- Builds large lookahead search tree with millions of positions
- State-of-the-art Go programs use MCTS

Convolutional Networks

- 12-layer CNN trained to predict expert moves
  - Raw CNN (looking at 1 position, no search at all) equals performance of MoGo with 105 position search tree

  1st strong Go program



Silver, et al., "Mastering the game of Go with deep neural networks and tree search," Nature, 2016.
https://deepmind.com/alphago/

Problems within RL

Learning and Planning

In sequential decision making

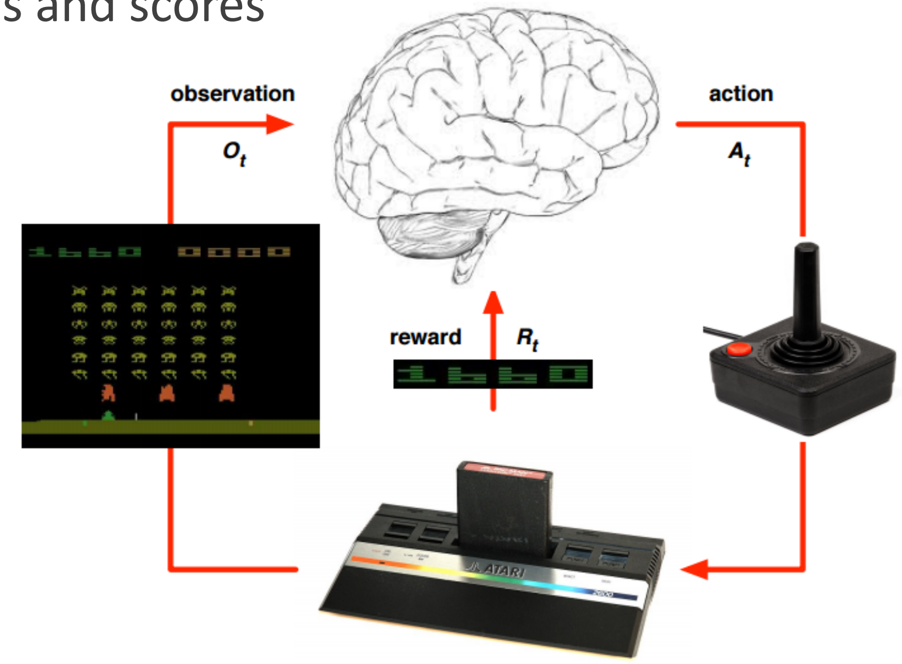
- Reinforcement learning
  - The environment is initially unknown
  - The agent interacts with the environment
  - The agent improves its policy
- Planning
  - A model of the environment is known
  - The agent performs computations with its model (w/o any external interaction)
  - The agent improves its policy (a.k.a. deliberation, reasoning, introspection, pondering, thought, search)

Atari Example: Reinforcement Learning

Rules of the game are unknown

Learn directly from interactive game-play

Pick actions on joystick, see pixels and scores



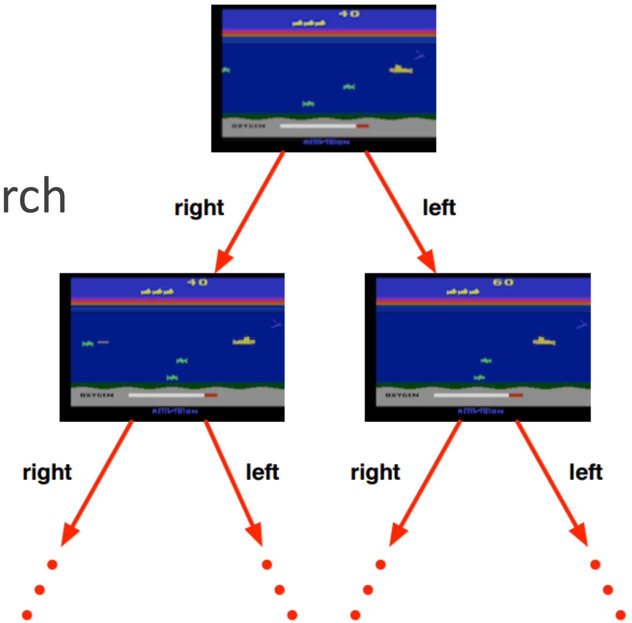
Atari Example: Planning

Rules of the game are known

Query emulator based on the perfect model inside agent's brain

- If I take action a from state s:
  - what would the next state be?
  - what would the score be?

Plan ahead to find optimal policy e.g. tree search



Exploration and Exploitation

Reinforcement learning is like trial-and-error learning

The agent should discover a good policy from the experience without losing too much reward along the way

When to try?

Exploration finds more information about the environment

Exploitation exploits known information to maximize reward

It is usually important to explore as well as exploit

RL for Unsupervised Model: **Modularizing Unsupervised Sense Embeddings (MUSE)**

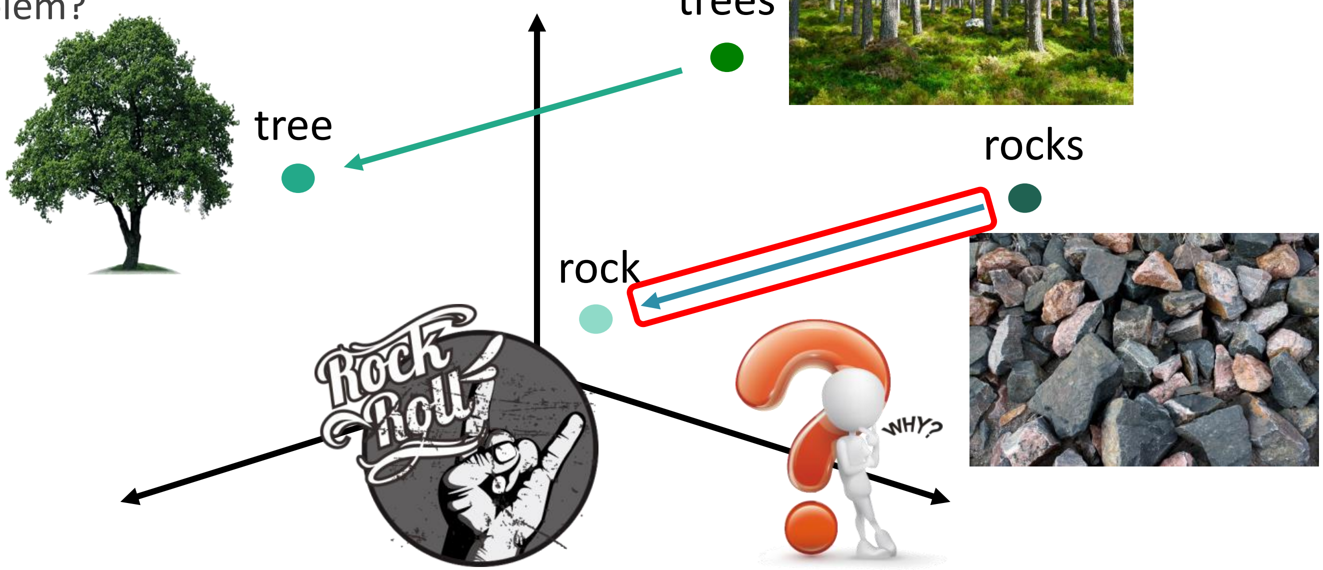
Word2Vec Polysemy Issue

Words are polysemy

- An apple a day, keeps the doctor away.
- Smartphone companies including apple, …

If words are polysemy, are their embeddings polysemy?

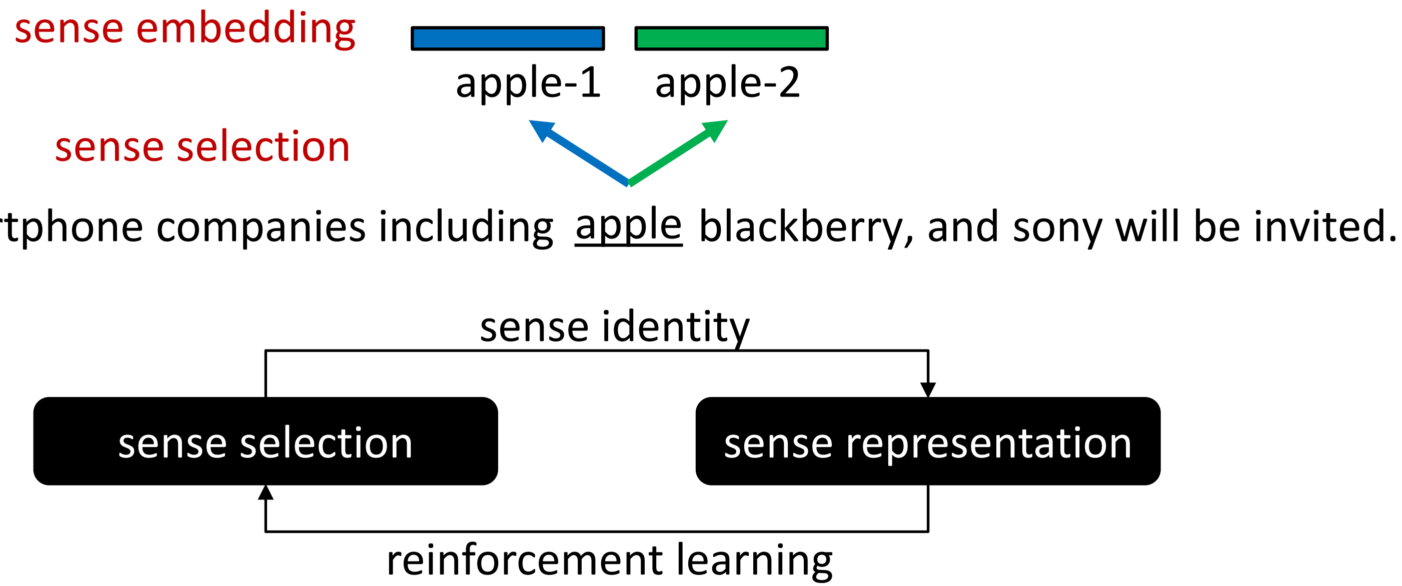
- No ☹
- What's the problem?



Modular Framework

Two key mechanisms

- **Sense selection** given a text context
- **Sense representation** to embed statistical characteristics of sense identity



Sense Selection Module

Input: a text context $\bar{C}_t = [C_{t-m}, \dots, C_t = w_i, \dots, C_{t+m}]$

Output: the fitness for each sense $z_{i1}, \dots, z_{i3}$

Model architecture: Continuous Bag-of-Words (CBOW) for efficiency

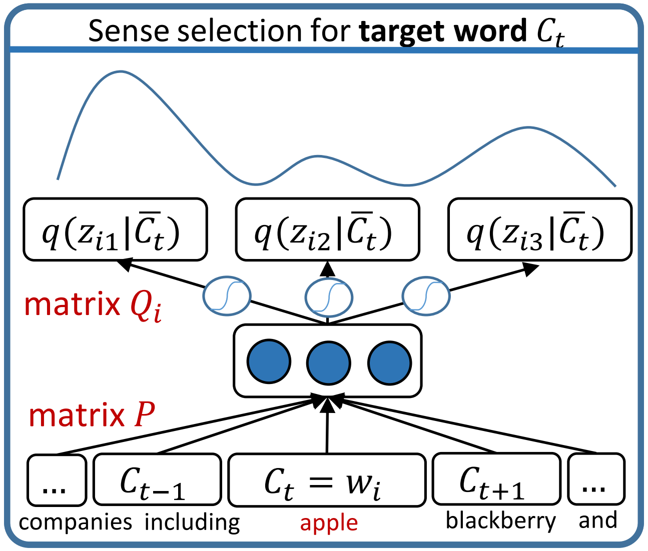
Sense selection

- Policy-based

$$\pi(z_{ik} \mid \bar{C}_t) = \frac{\exp(Q_{ik}^T \sum_{j \in \bar{C}_t} P_j)}{\sum_{k' \in Z_i} \exp(Q_{ik'}^T \sum_{j \in \bar{C}_t} P_j)}$$

- Value-based (Q-value)

$$q(z_{ik} \mid \bar{C}_t) = \sigma(Q_{ik}^T \sum_{j \in \bar{C}_t} P_j)$$



Sense Selection Module

Sense Representation Module

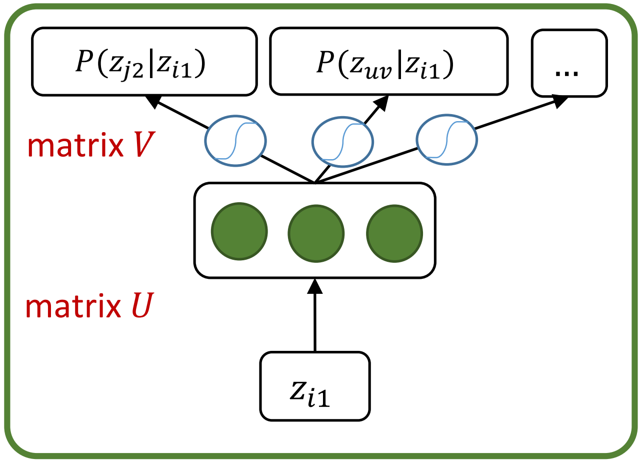
Input: sense collocation z_{ik}, z_{jl}

Output: collocation likelihood estimation

Model architecture: skip-gram architecture

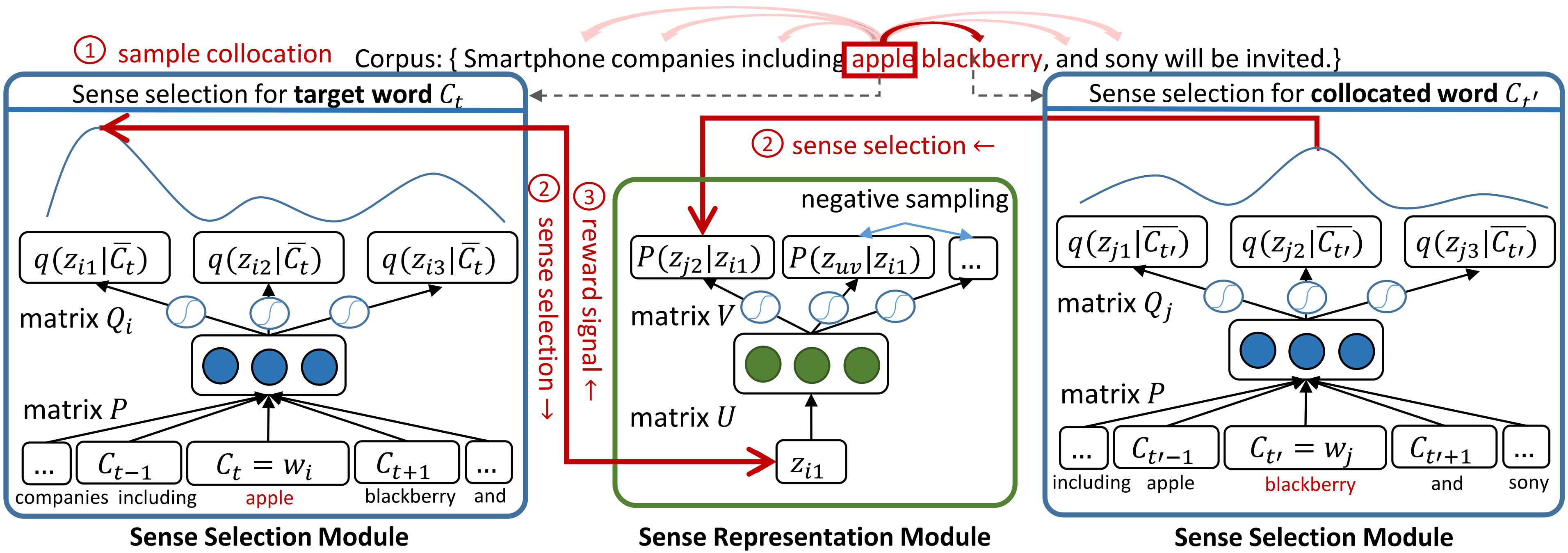
Sense representation learning

$$\log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) = \log \sigma(U_{z_{ik}}^T V_{z_{jl}}) + \sum_{v=1}^{M} \mathbb{E}_{z_{uv} \sim p_{neg}(z)}[\log \sigma(-U_{z_{ik}}^T V_{z_{uv}})]$$



Sense Representation Module

A Summary of MUSE



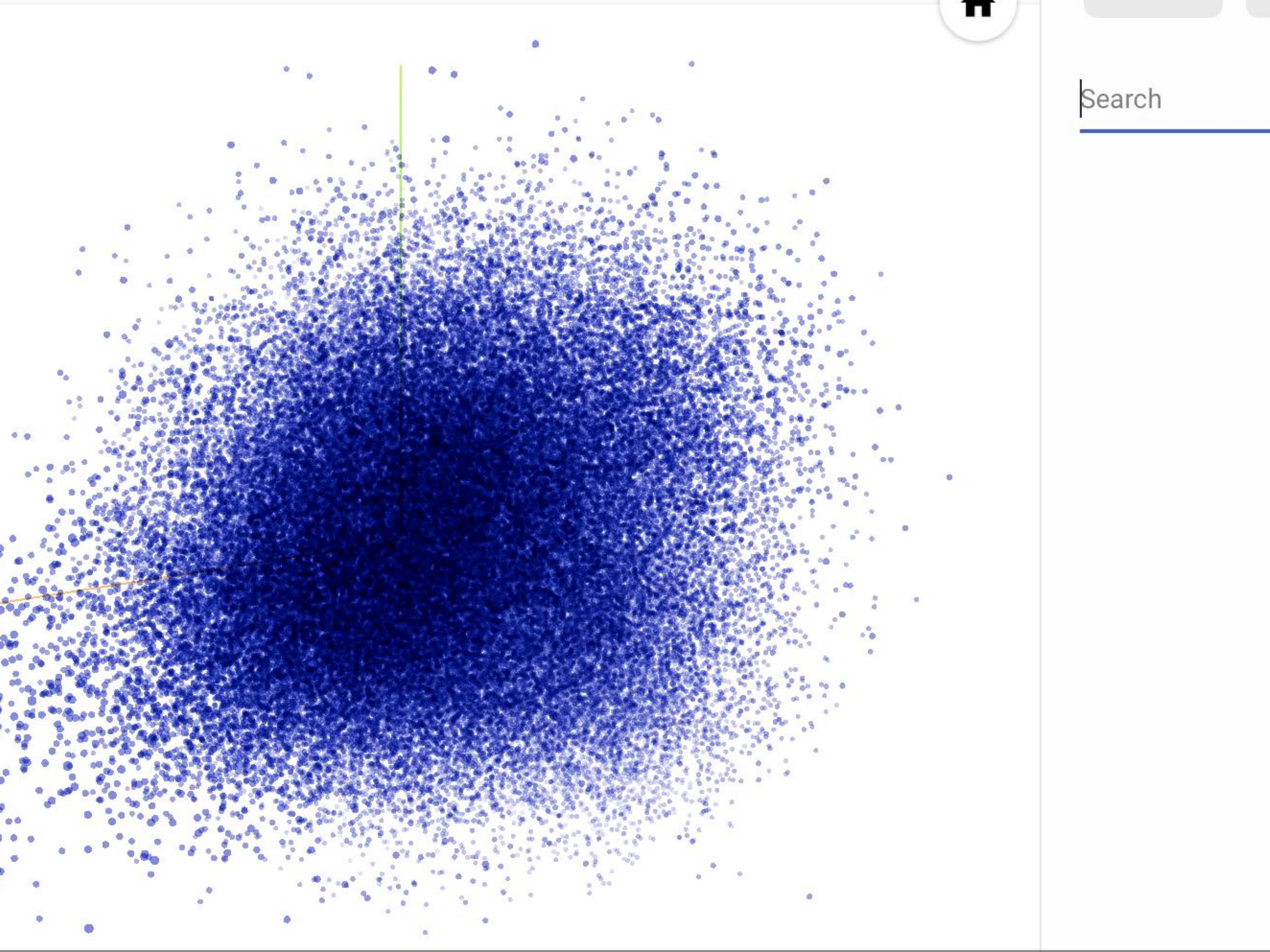
The first purely sense-level embedding learning with efficient sense selection.

Qualitative Analysis

Context	… braves finish the season in **tie** with the los angeles dodgers …	… his later years proudly wore **tie** with the chinese characters for …
k-NN	scoreless otl shootout 6-6 hingis 3-3 7-7 0-0	pants trousers shirt juventus blazer socks anfield
Figure		

Qualitative Analysis

Context	… of the mulberry or the **blackberry** and minos sent him to …	… of the large number of **blackberry** users in the us federal …
k-NN	cranberries maple vaccinium apricot apple	smartphones sap microsoft ipv6 smartphone
Figure		


Search

OpenAI Universe

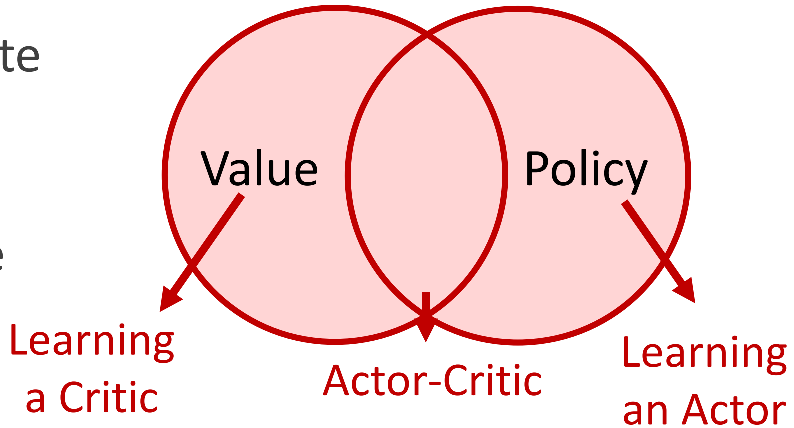
Software platform for measuring and training an AI's general intelligence via the OpenAI gym environment

Concluding Remarks

RL is a general purpose framework for **decision making** under interactions between agent and environment

An RL agent may include one or more of these components

- Value function: how good is each state and/or action
- Policy: agent's behavior function
- Model: agent's representation of the environment



RL problems can be solved by end-to-end deep learning

Reinforcement Learning + Deep Learning = AI