Question Answering Systems

Reinforcement learning in QA

Rishiraj Saha Roy
Max Planck Institute for Informatics, Germany

10

Saarland University, Summer Semester 2020

Question of the day

RL

How can we use reinforcement learning to improve QA systems?

or build-from-scratch

You'll find this covered in

- Ask the Right Questions: Active Question Reformulation with Reinforcement Learning
  - Buck et al.
  - ICLR 2018
  - https://openreview.net/pdf?id=S1CChZ-CZ

Google

- Go for a Walk and Arrive at the Answer: Reasoning over Paths in Knowledge Bases using Reinforcement Learning

KGR

  - Das et al.
  - ICLR 2018
  - https://openreview.net/pdf?id=Syg-YfWCW

Learning in QA

1 ▪ Unsupervised/weakly supervised learning

- QUEST, PARALEX, ... Ravichandran & Hovy 2002

2 ▪ Supervised learning

- AQQU, STAGG, SEMPRE, ...

3 ▪ Reinforcement learning: Why?

- AQA, MINERVA, ...

Active QA

Alternative:
- Online learning
- Regression
- Distant supervision

more natural: RL framework

- No training data
- Reward → noisy signal towards correctness

↳ 0.623 → +1/−1 ↳ signal for SL

− 0.121 ↳ not clean labels

model

Reinforcement learning: Basics

1 ▪ States

2 ▪ Actions — Take step fwd/ R/L/B; Stay still

3 ▪ Policy — strategy

4 ▪ Reward

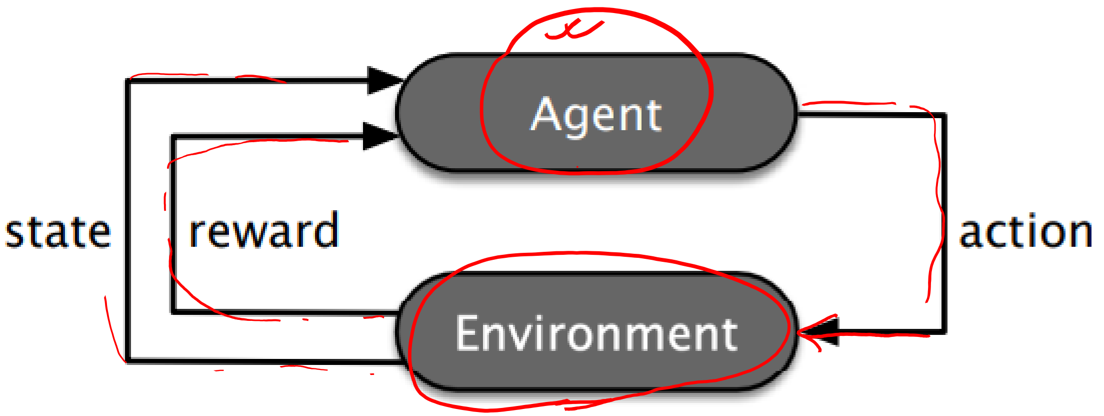
5 ▪ Value → long term reward
function

6 ▪ Environment model

Tumble Poor Out

state Inputs

wall



Robot: go out of a room
Why RL?
- Lot of training data
trial & error

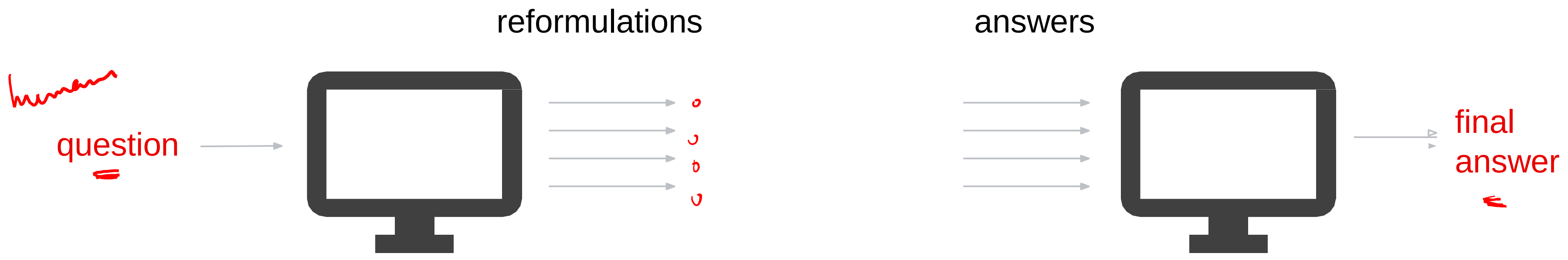
Games
Chess

Research paper 1

Ask the Right Questions: Active Question Reformulation with Reinforcement Learning

Basic idea

correct / satisfactory

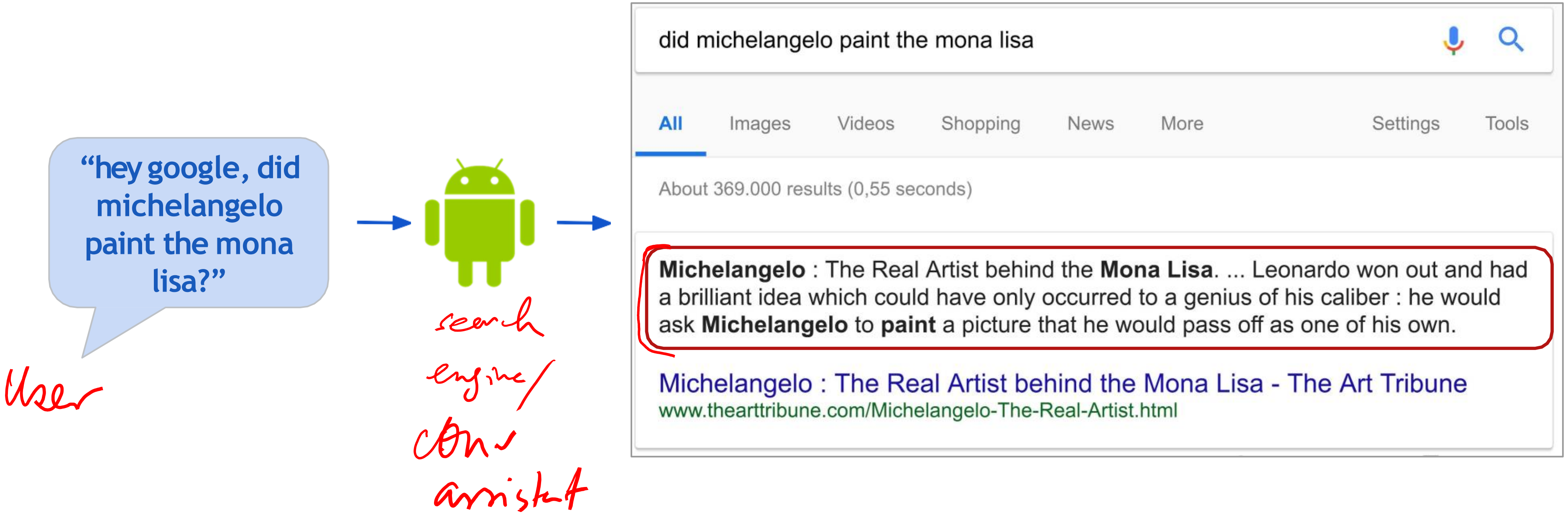


- Train an agent that learns to ask 'optimized' questions
- Machine learns a non-trivial, non-human, but interesting policy to generate new question
- First step towards an interactive language agent

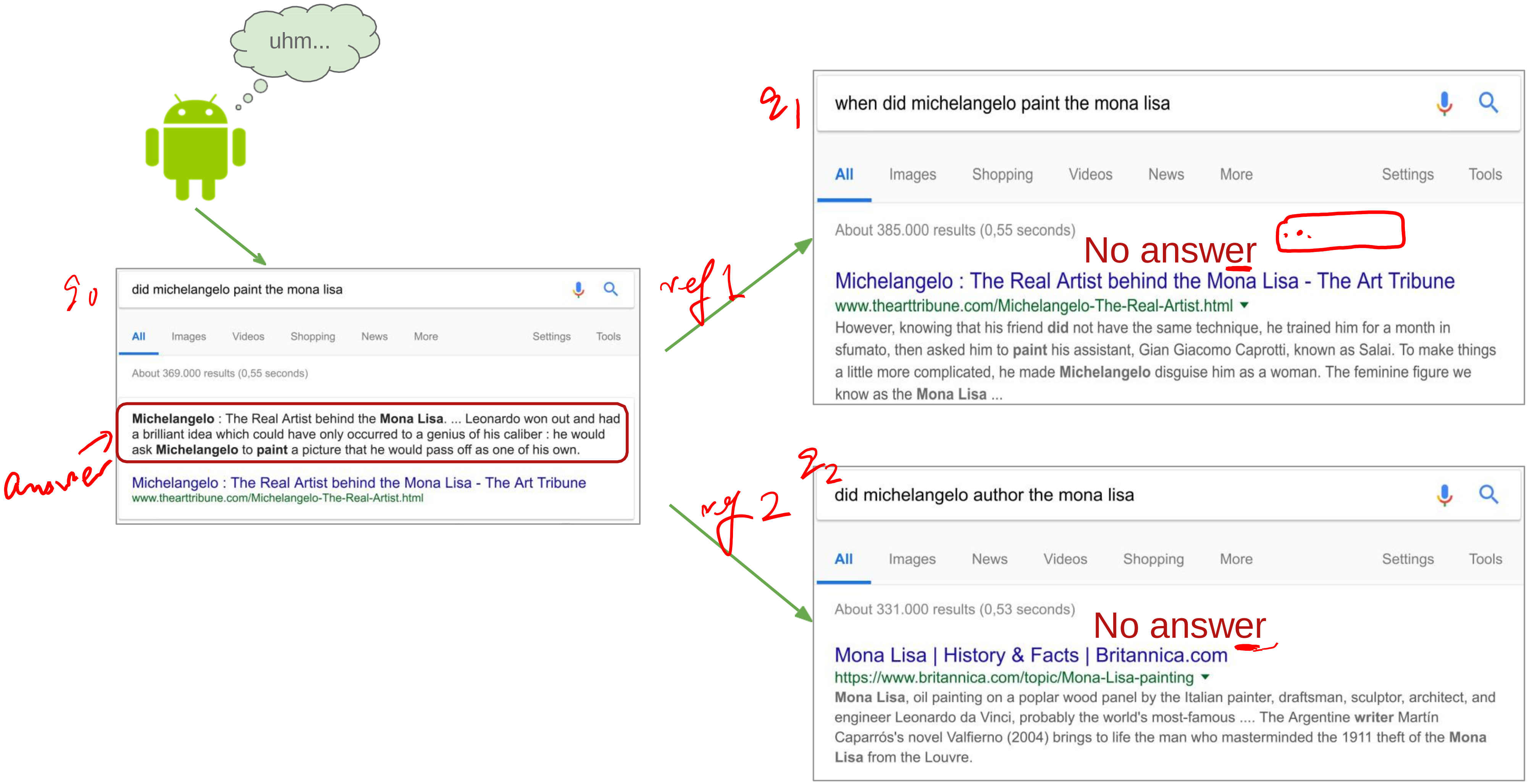
Thanks to Christian Buck for the slides

Intelligent language agent for QA

Vision

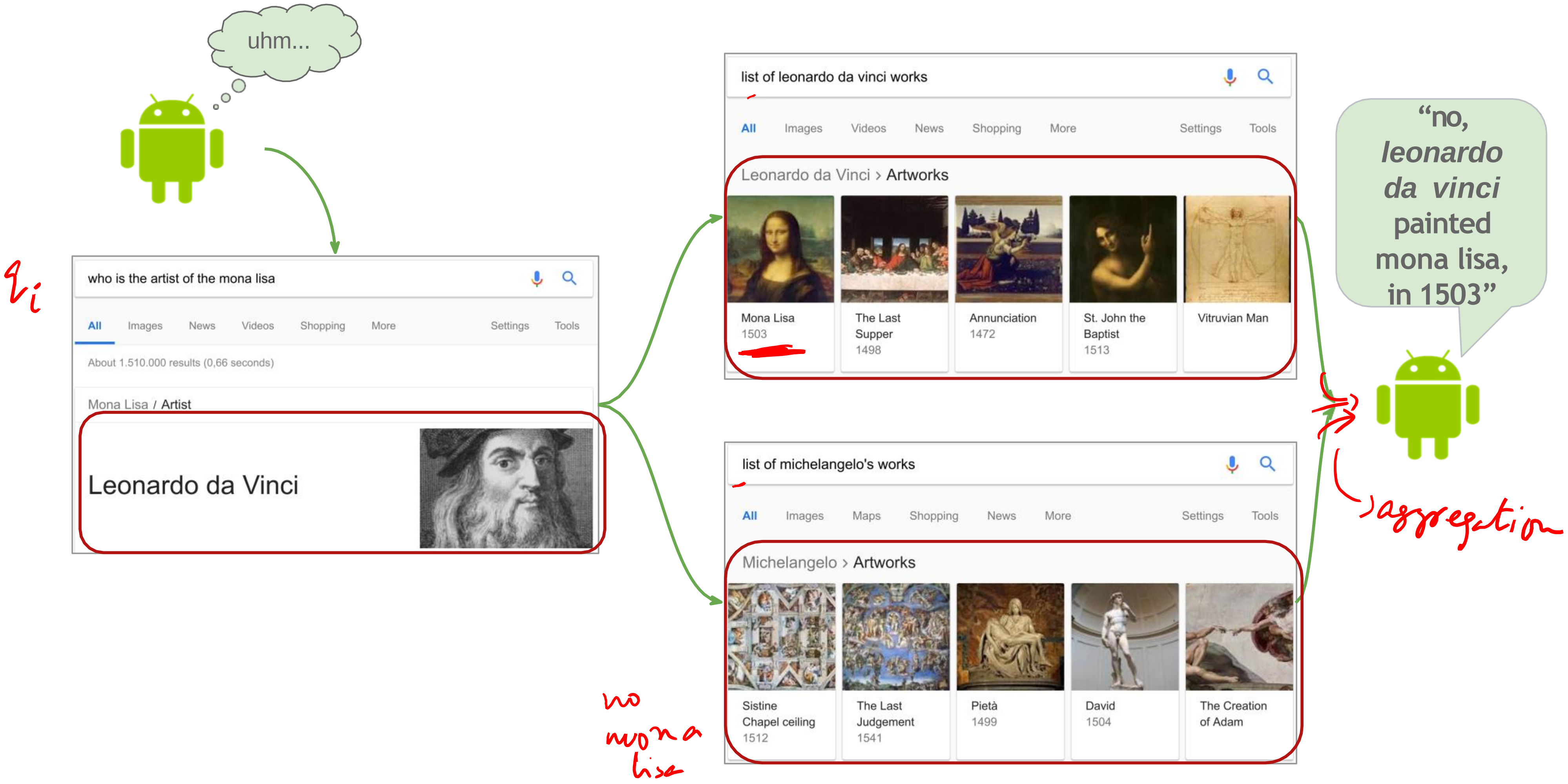


Intelligent language agent for QA



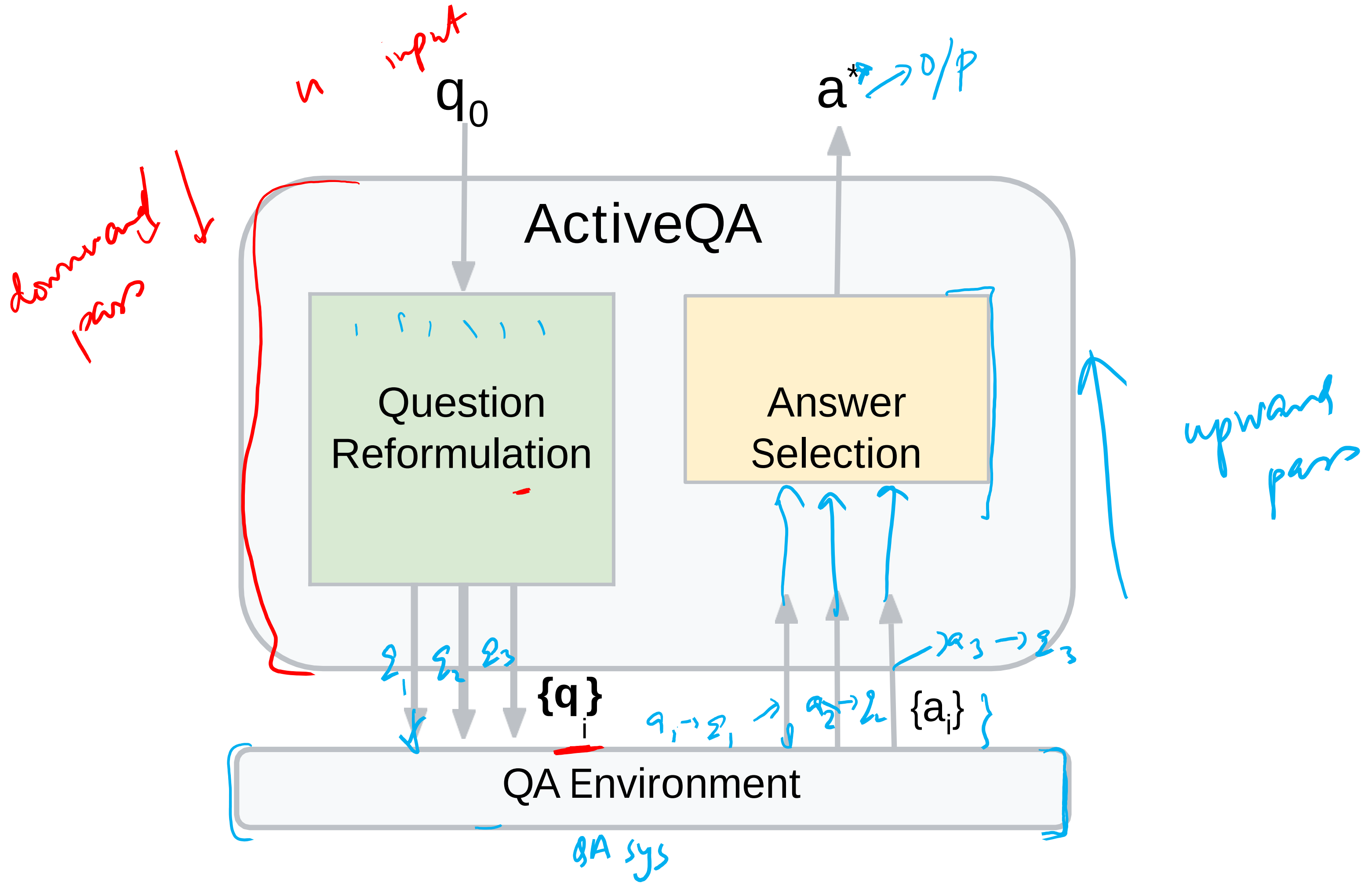
Intelligent language agent for QA

Vision



AQA: Active question answering

Active QA



Seq-to-seq

Active QA: Sequence-to-sequence

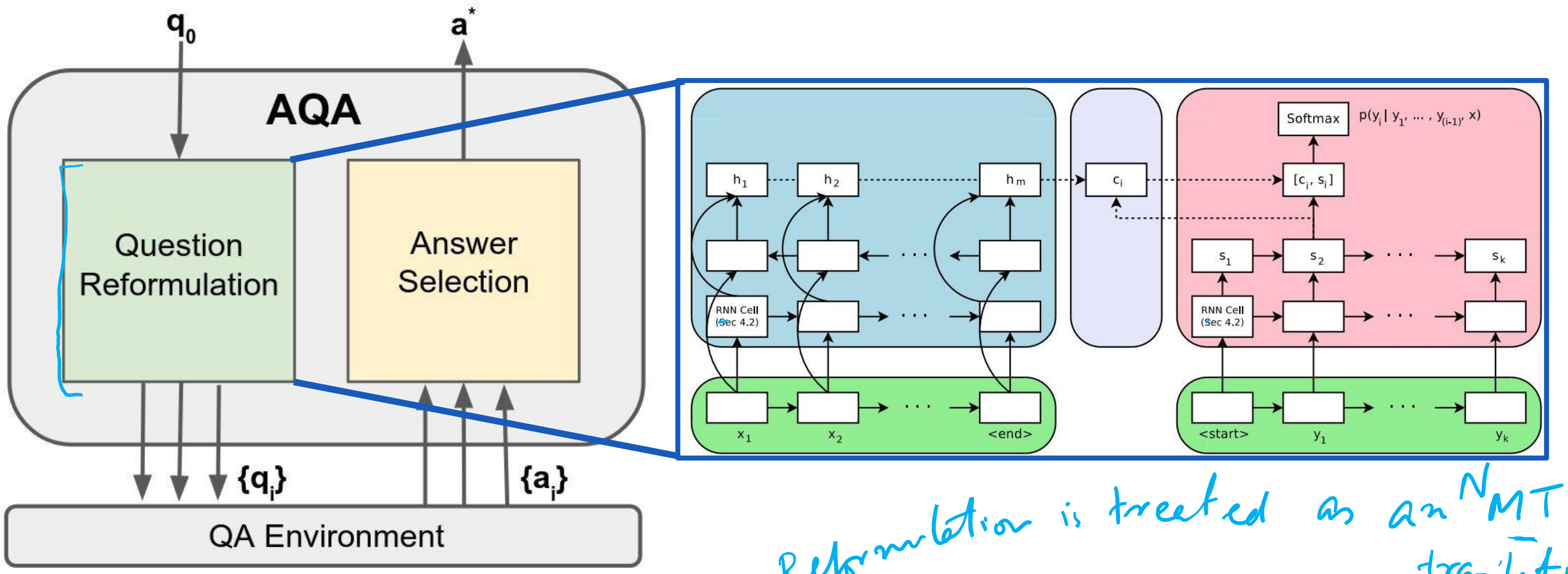


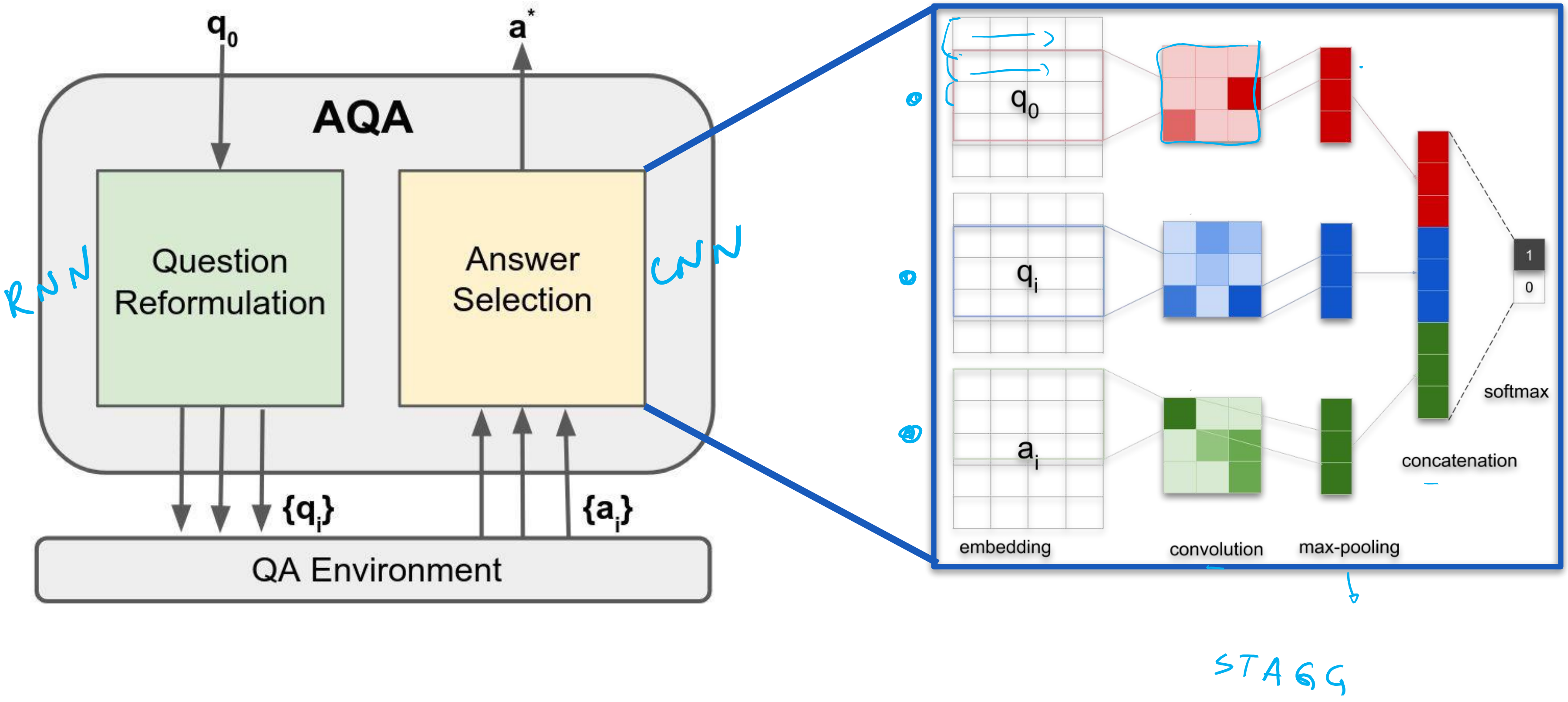
*Image from [Britz et al., 2017]

Reformulation is treated as an NMT task
translation

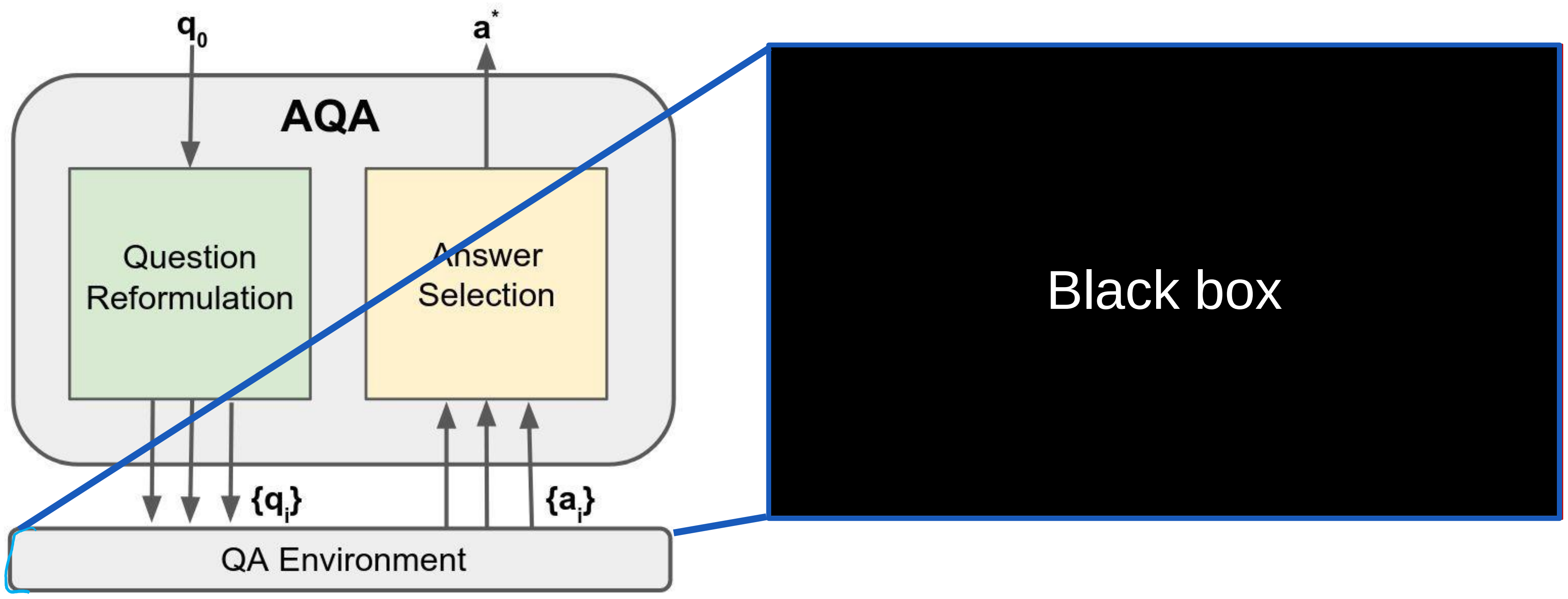
ooo ENG → GER ooo
ENG ← GER
user's lang q_0 ↔ "m/c" lang

birthplace of AE
where was AE born

Active QA: Answer selection



Active QA: Answer selection



Search engine

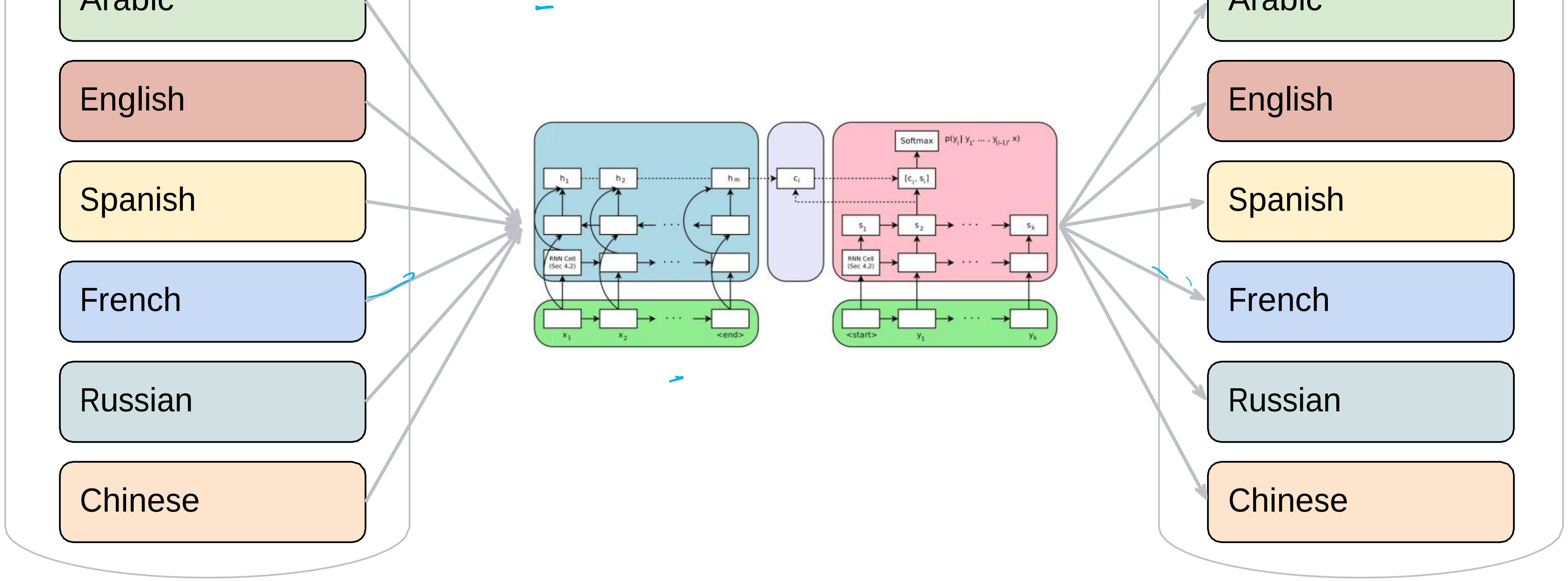
Training

- Supervised paraphrasing
- Question reformulation with reinforcement learning
- Answer selection

Training 1: Initialization with supervised paraphrasing model

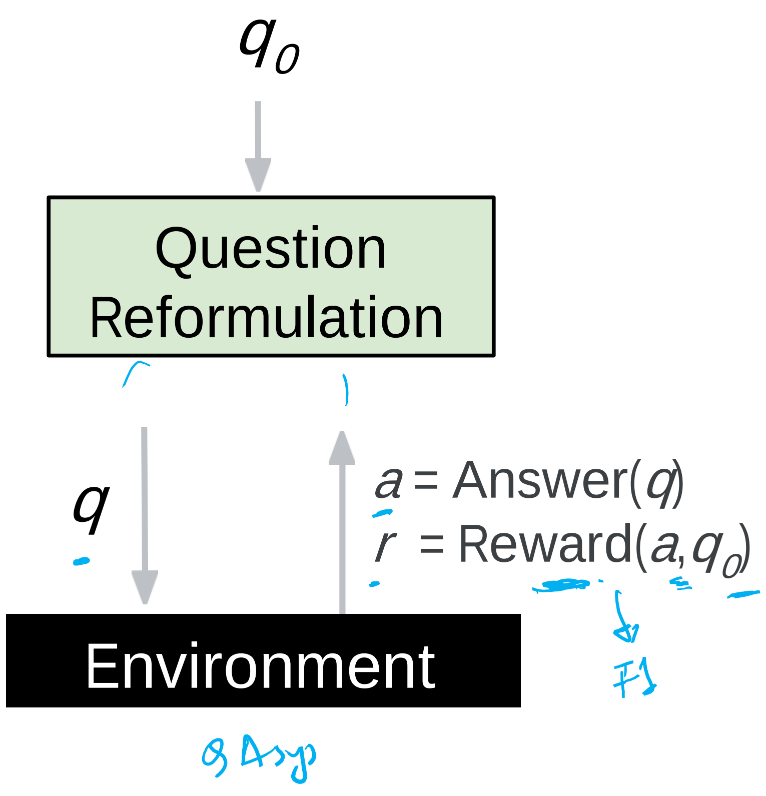
→ German → English

1. 300M multilingual pairs "zero-shot"
2. 1M English paraphrases



Training 2: Reformulation model

RL



policy network
parameterized policy

Use policy gradient to maximize expected reward:

$$\mathbb{E}_{p(q|q_0)} \text{Reward}(a, q_0)$$

REINFORCE
Williams 1992
" & Peng 1991

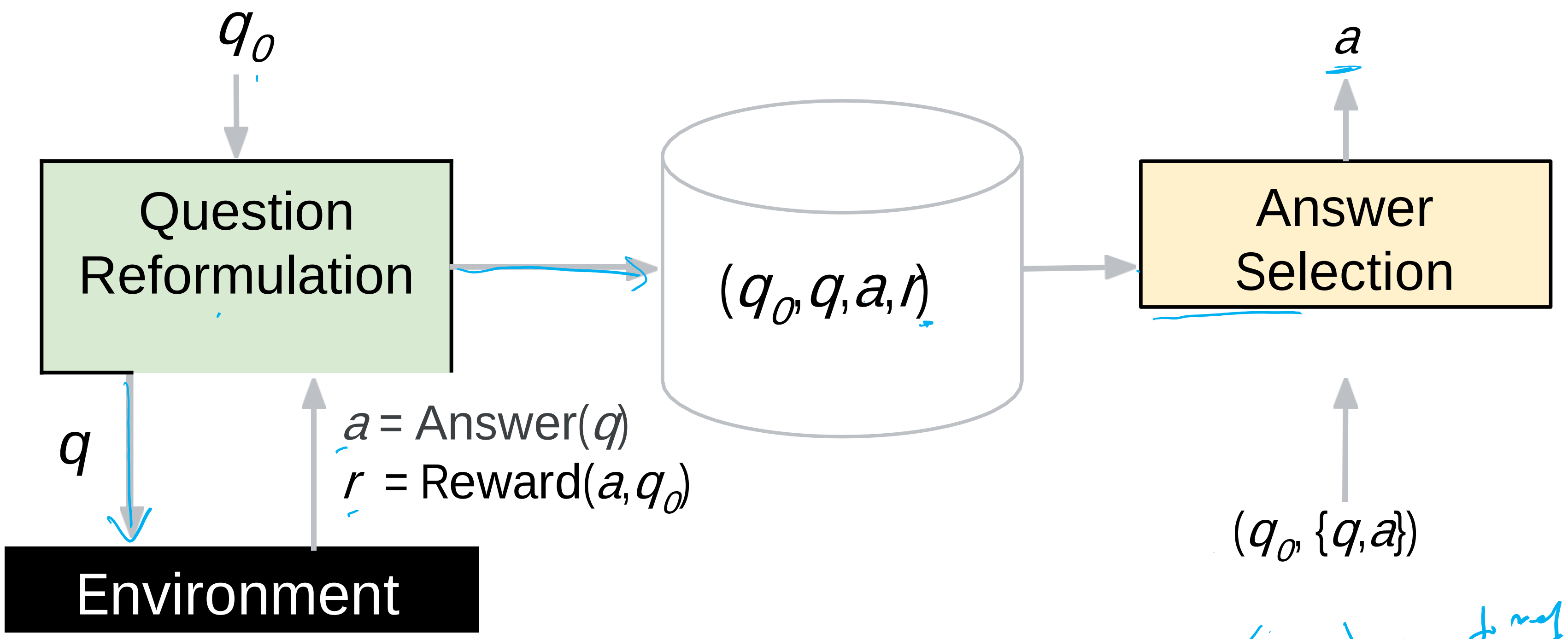
REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ $\quad (G_t)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Training 3: Answer selection

Supervised training on tuples scraped during reformulator training



supervised sys. with CNN

ref I am doing

SearchQA: Reading comprehension

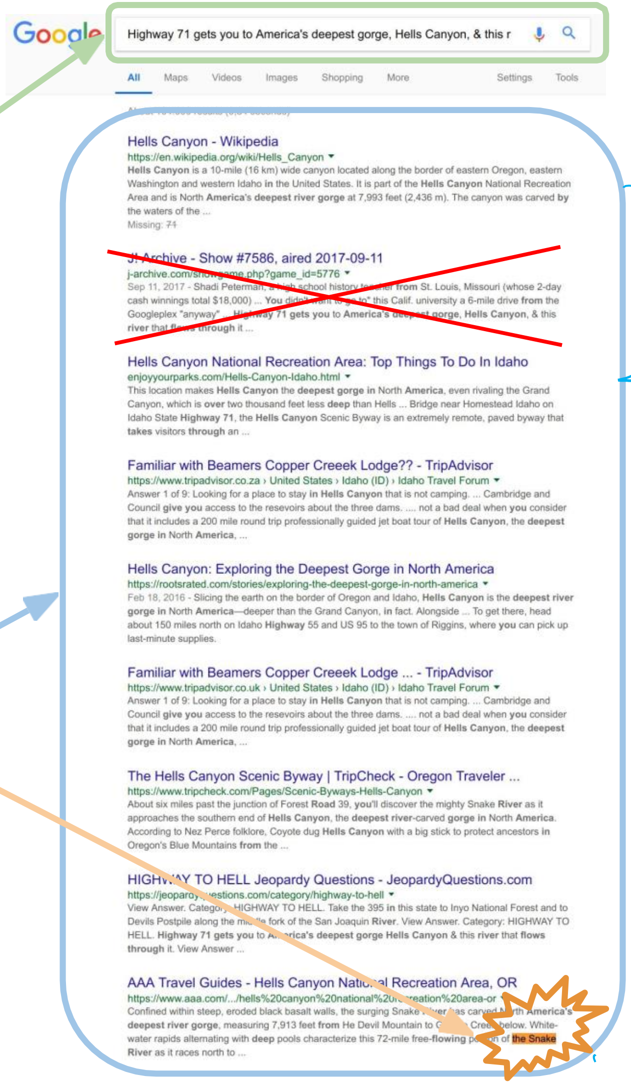
MRC [passage/context]

trivia

Question (J! *clue*): “Highway 71 gets you to America's deepest gorge, Hells Canyon, & this river that flows through it”

Answer: the Snake

Context: Top 50-100 search snippets for the question filtered for giveaways



[Dunn et. al, 2017]

ActiveQA: Environment

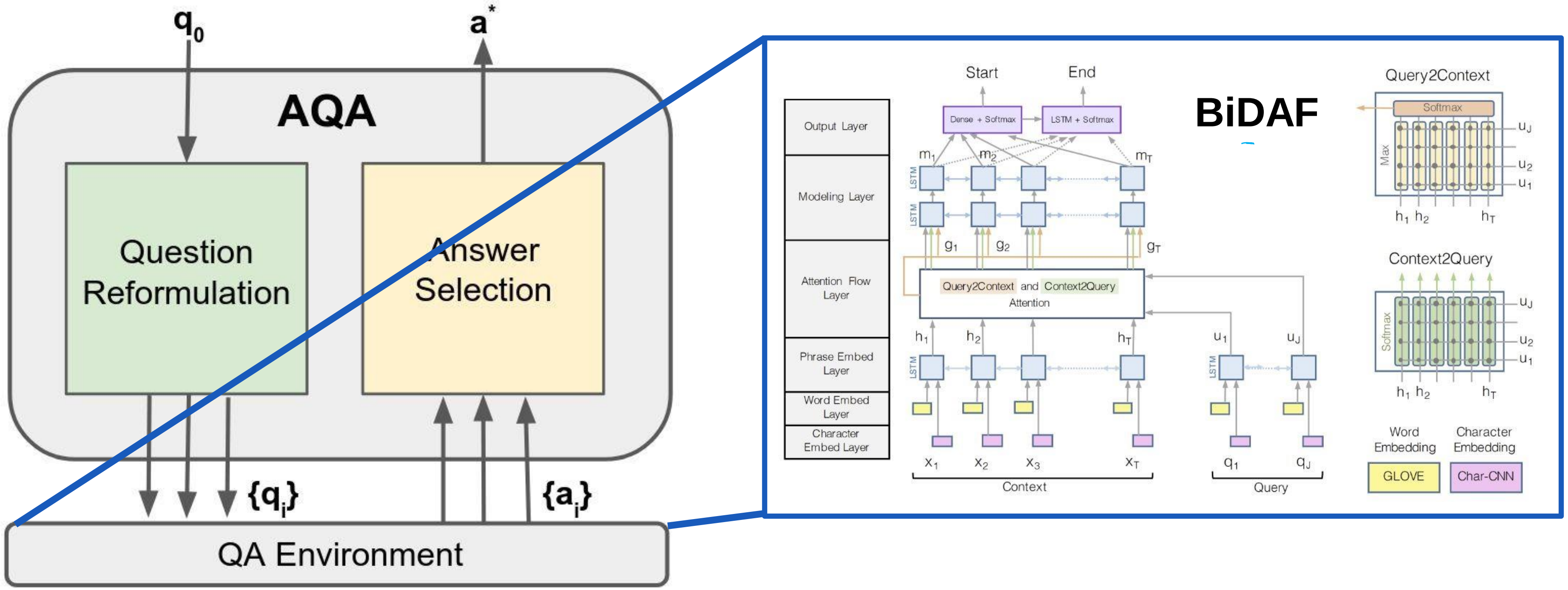
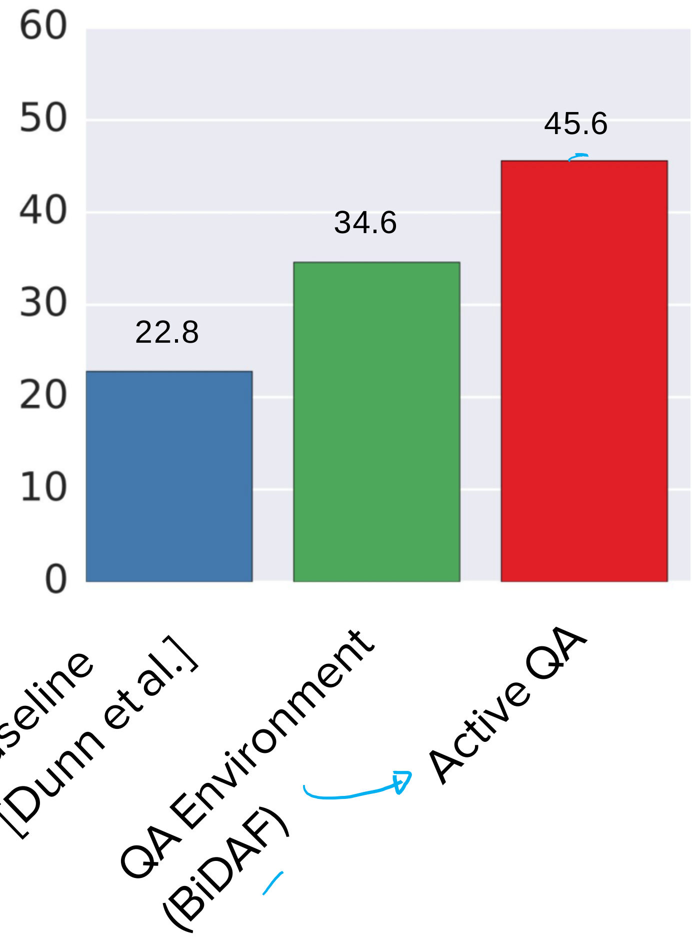


*Image from [Seo et al. 2017]

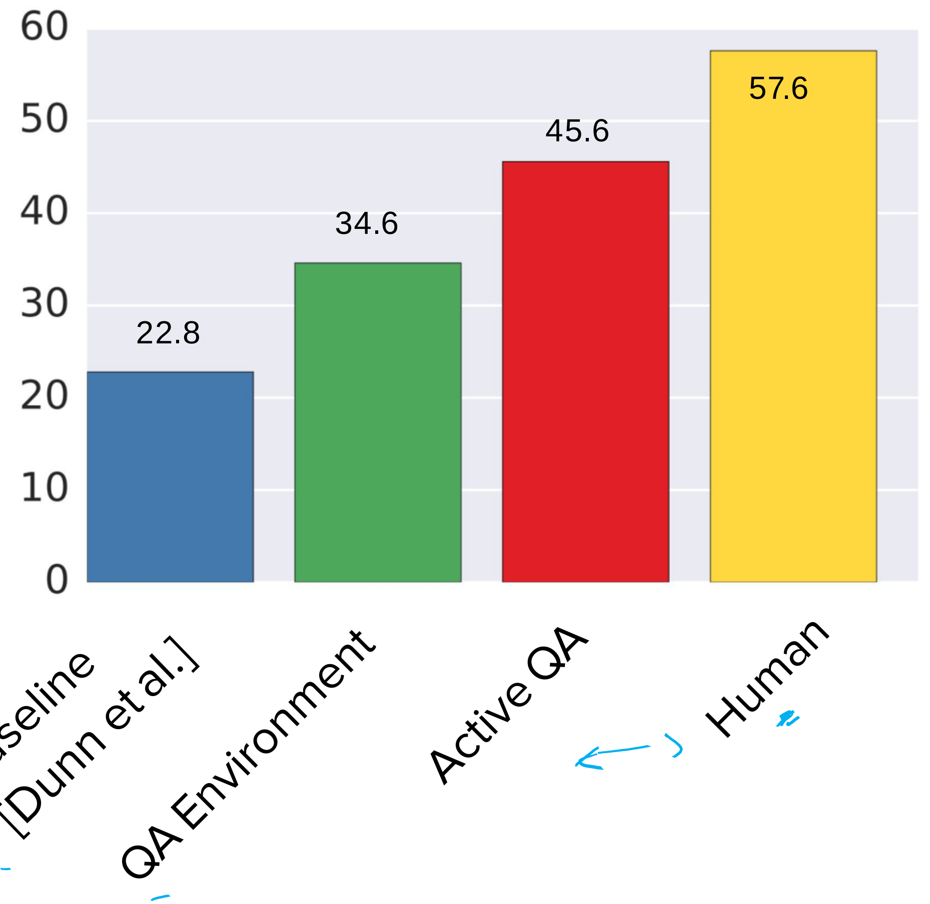
Results

Answer quality: F1 Score

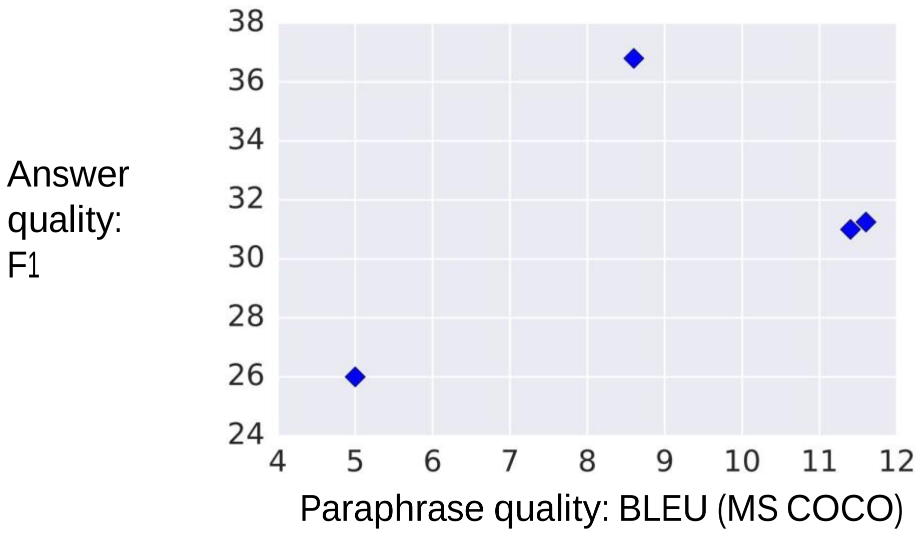


Results

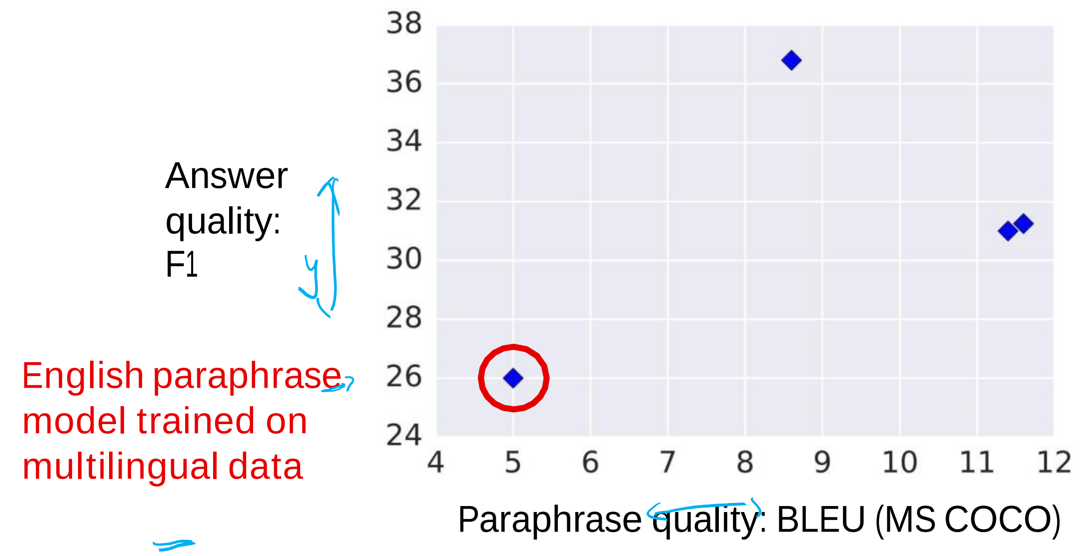
Answer quality: F1 Score



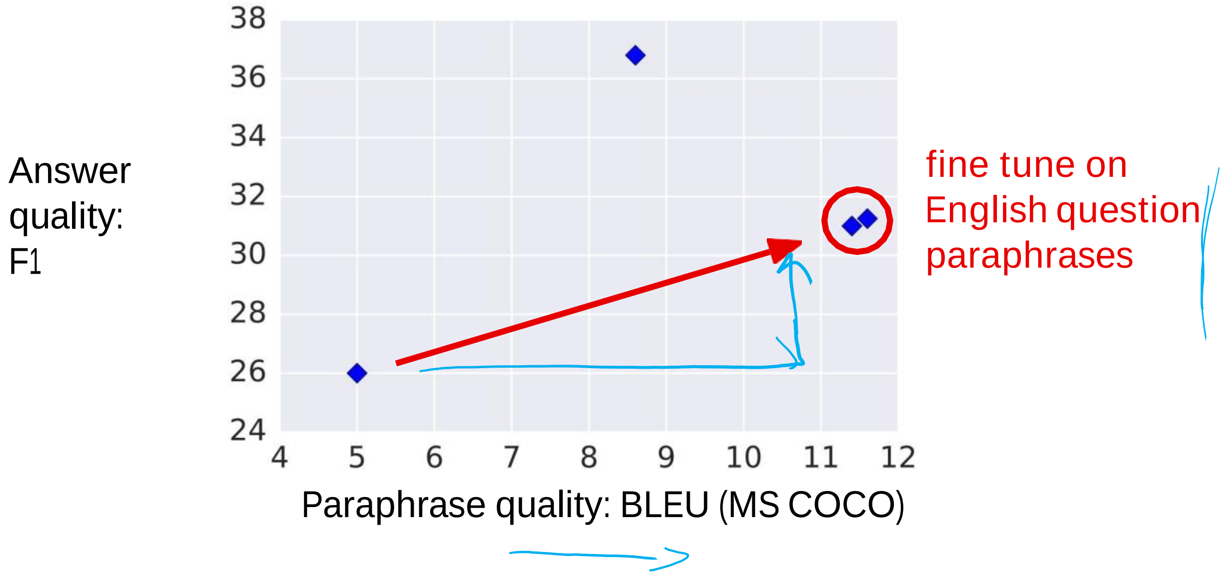
Reformulation quality



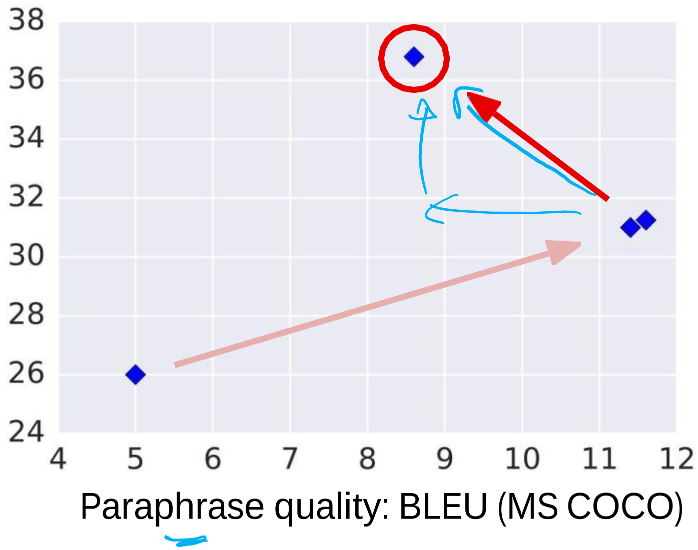
Reformulation quality



Reformulation quality



Reformulation quality

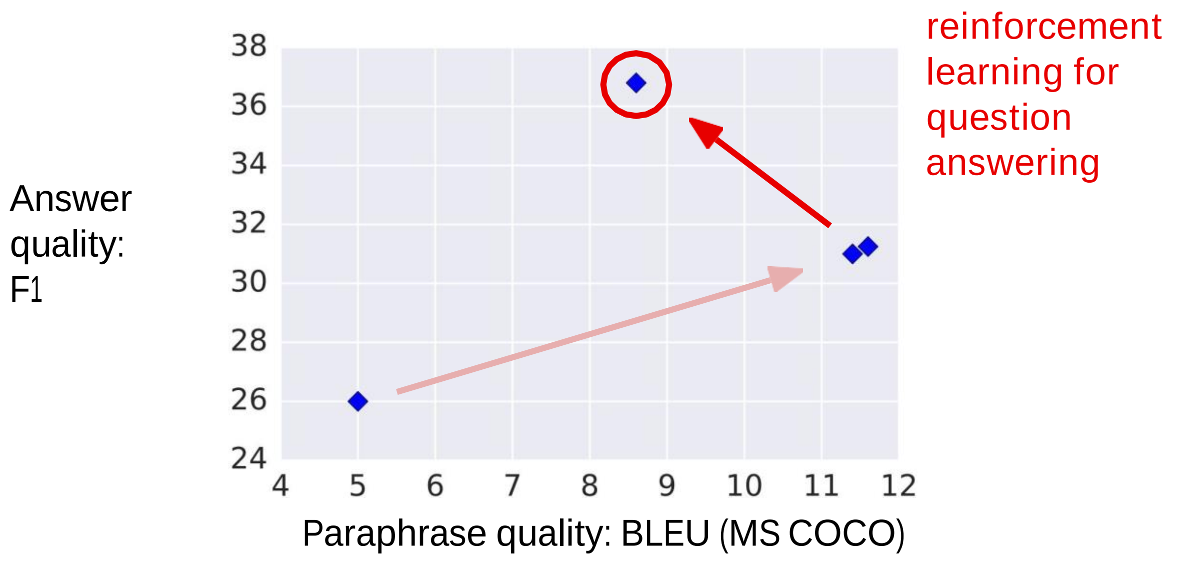


Answer quality: F1

Paraphrase quality: BLEU (MS COCO)

reinforcement learning for question answering

Reformulation quality



- Input: type humorous poem bears name irish port city
- Top reformulation: what is name humorous poem poem bears city city city

Research paper 2

Go for a Walk and Arrive at the Answer:

Reasoning over Paths in Knowledge Bases

using Reinforcement Learning

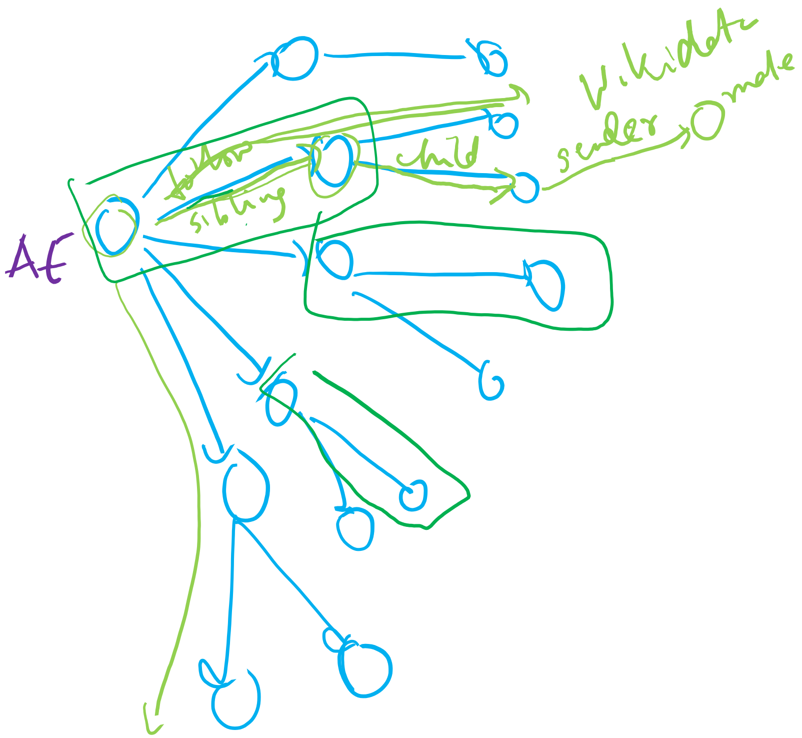
MINERVA: Task in KG Reasoning

I/p : relation + start entity

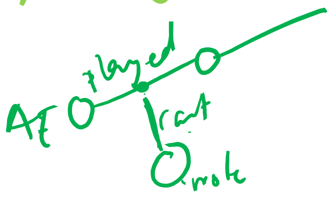
nephew

AE

O/p: best path



no qualifiers



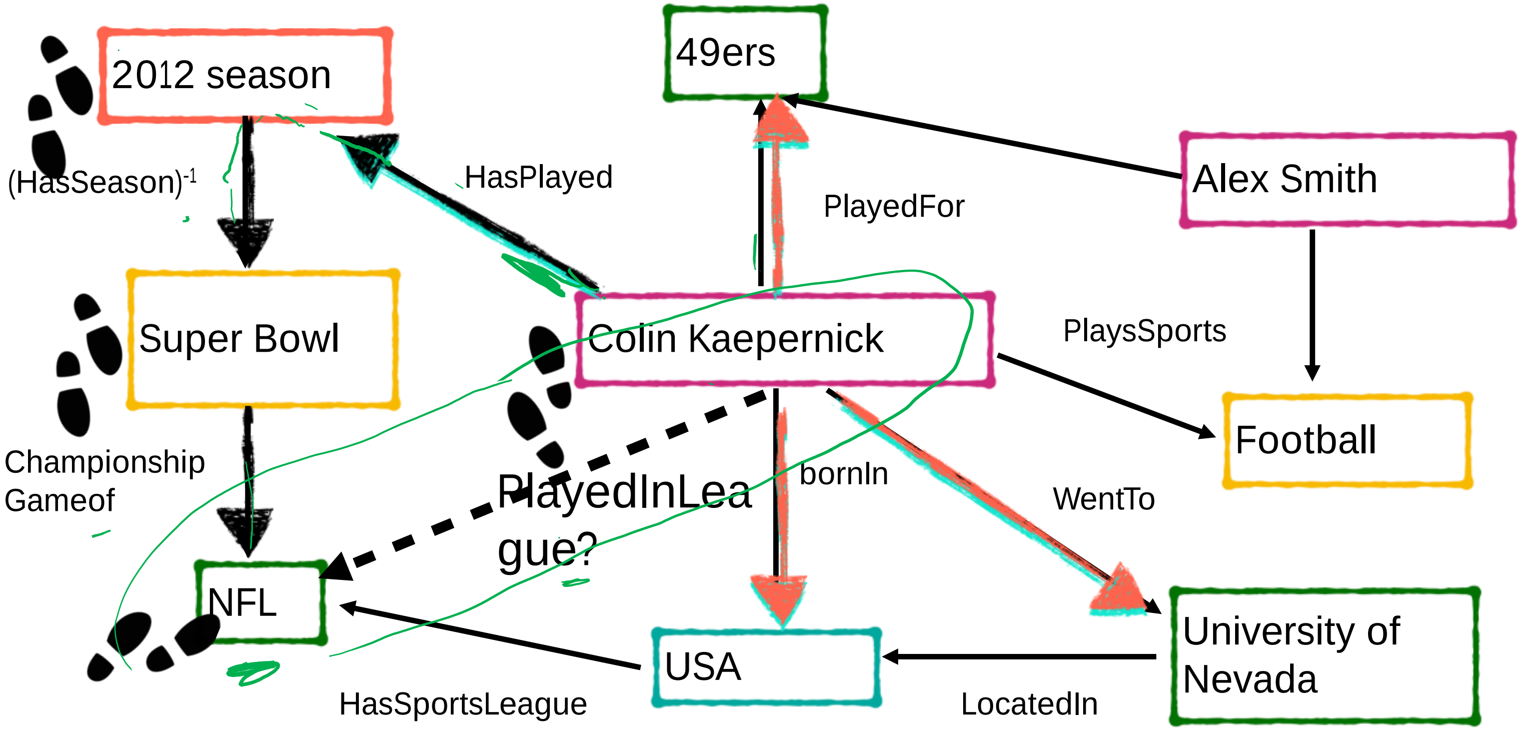
- KGR ~ QA

SE → NERD

formulation

MINERVA: Task

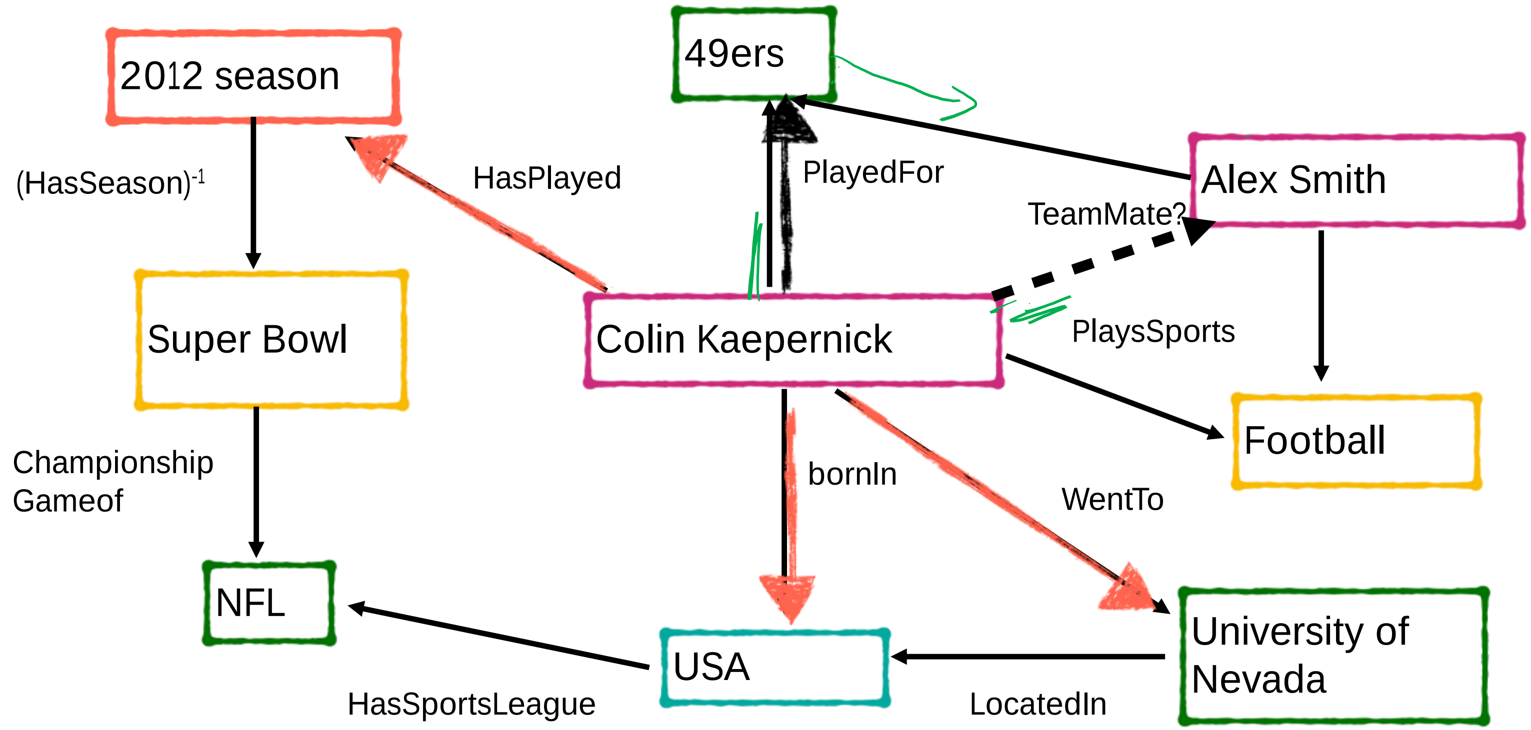
Meandering thru Networks of Entities to reach Verisimilar Answers?



question :? → (Colin Kaepernick, PlayedInLeague, ?)
rel

Thanks to
Rajarshi Das
for the slides

MINERVA: Task

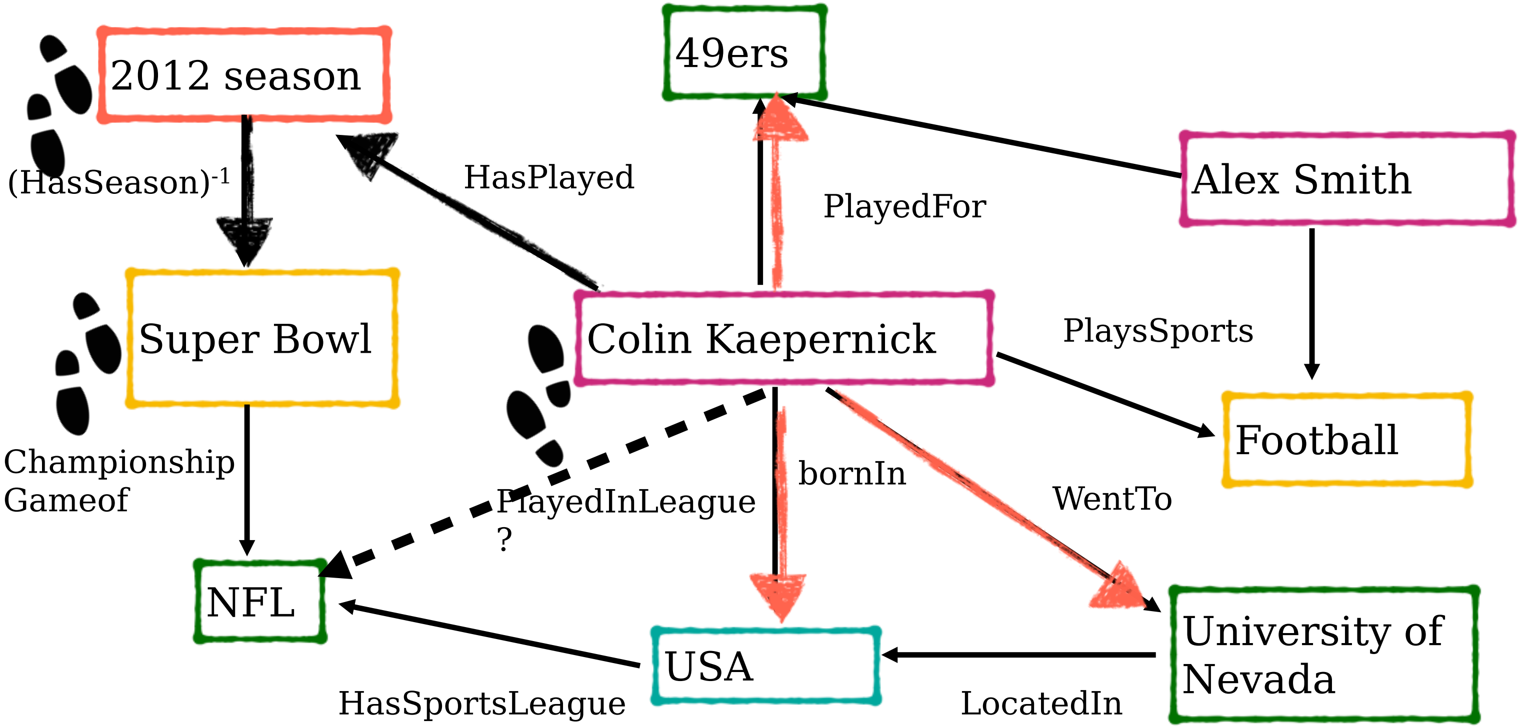


Typ 1: (Colin Kaepernick, PlayedInLeague, ?)

Typ 2 → (Colin Kaepernick, TeamMate/Co-Worker, ?)

Query Dependent
Decision Making!

MINERVA: Model



(Colin Kaepernick, PlayedInLeague, ?)

MINERVA: Model

- Input - (Colin Kaepernick, PlayedInLeague, ?)

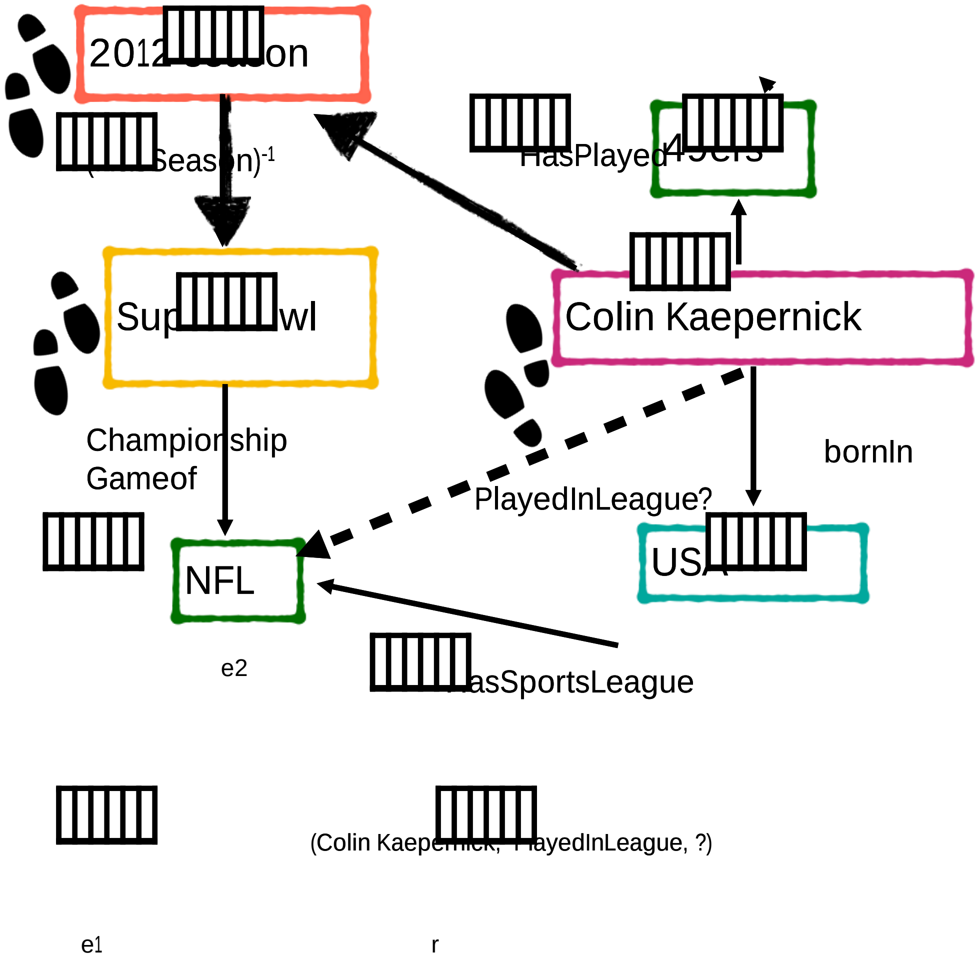
Partially Observed Markov decision process

POMDP

4-tuple

States - (e_t, e_1, r, e_2)

Not observed by the agent



MINERVA: Policy

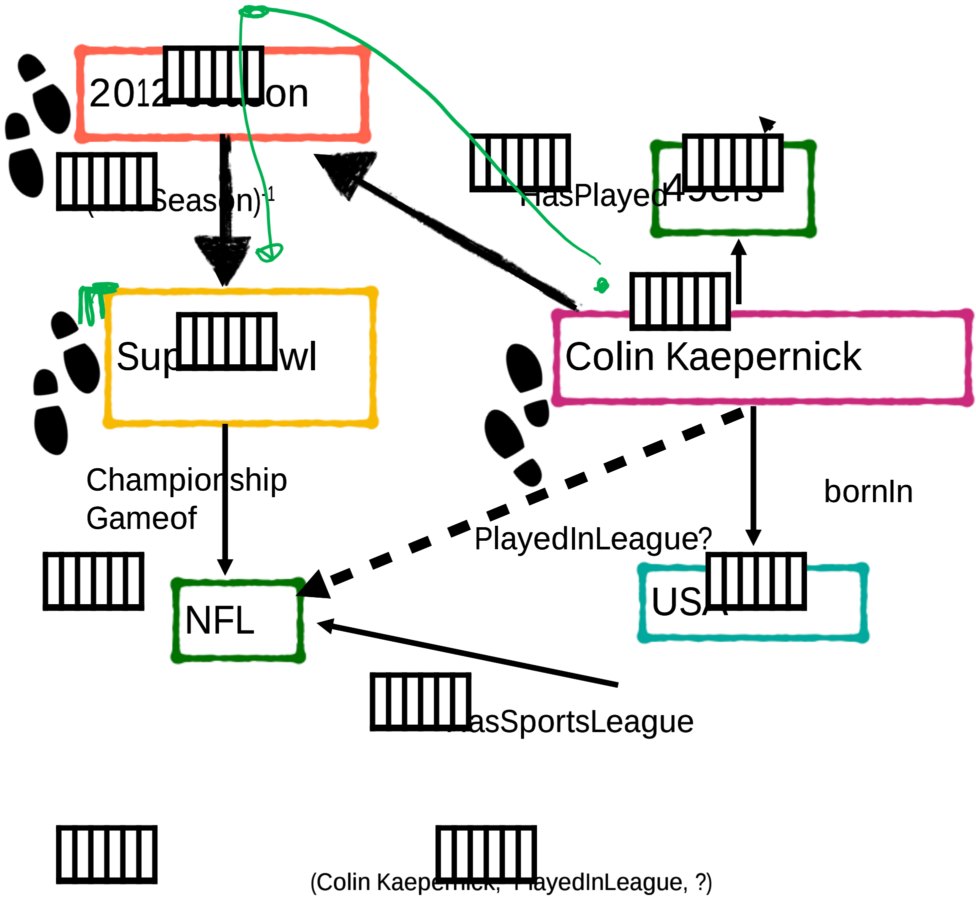
- Input - (Colin Kaepernick, PlayedInLeague, ?)

Partially Observed Markov decision process

States - (e_t, e_1, r, e_2)

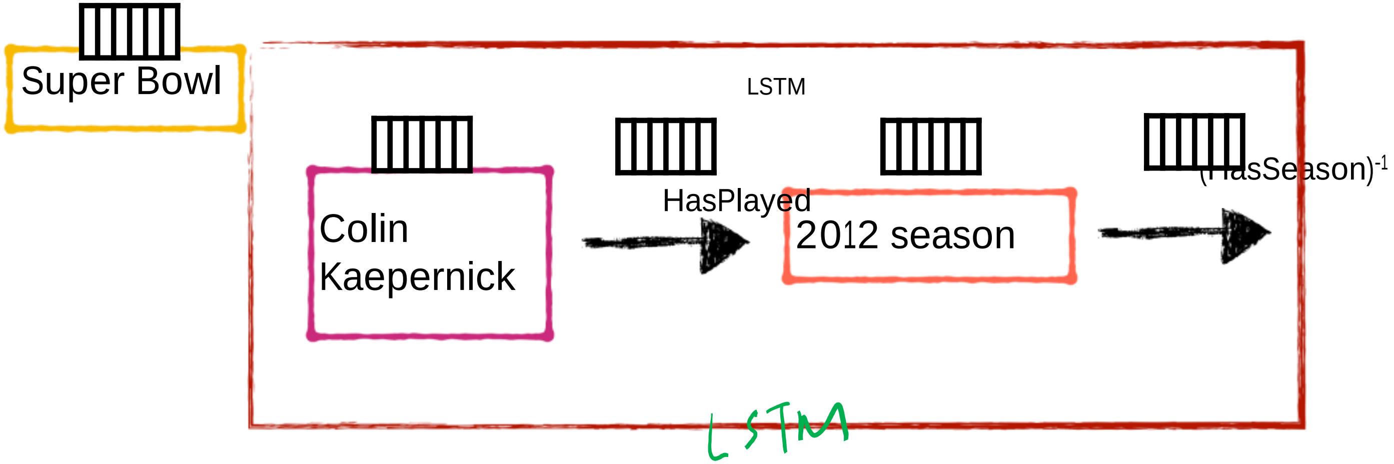
Policy

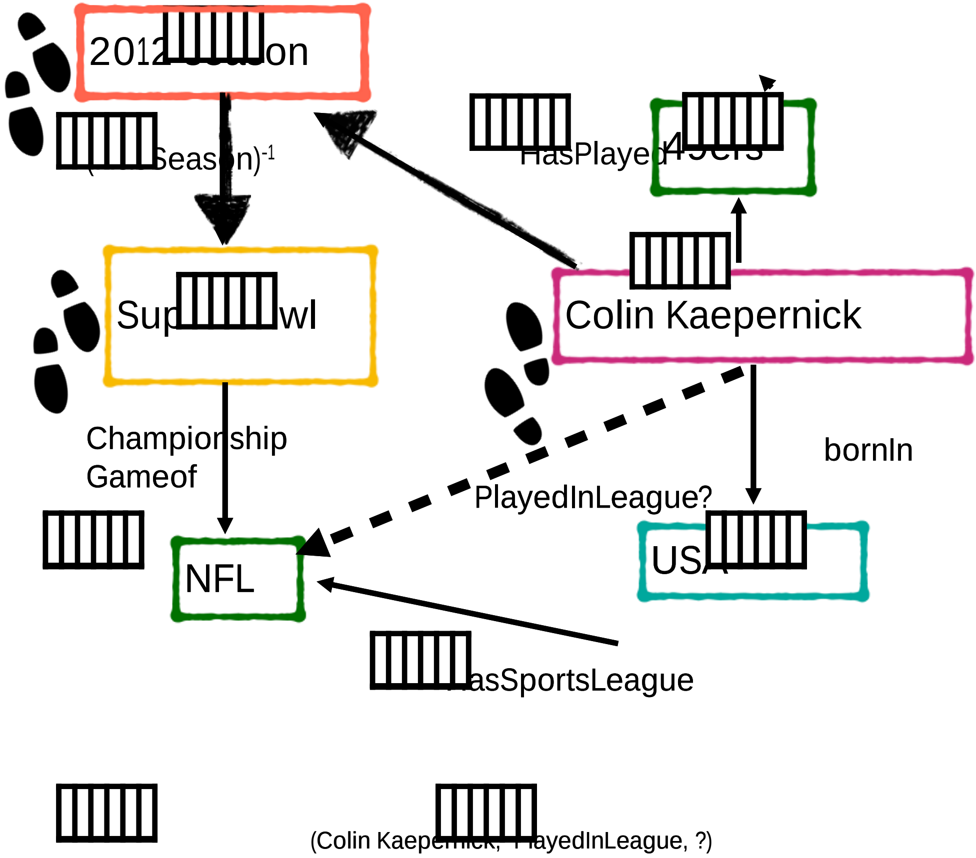
- Randomized & *history* dependent



MINERVA: Policy

- History



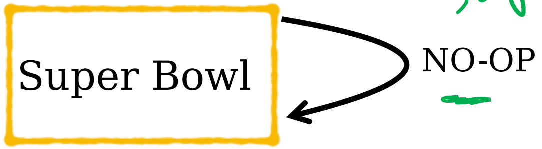


MINERVA: Actions

- All possible outgoing edges

Actions of agent

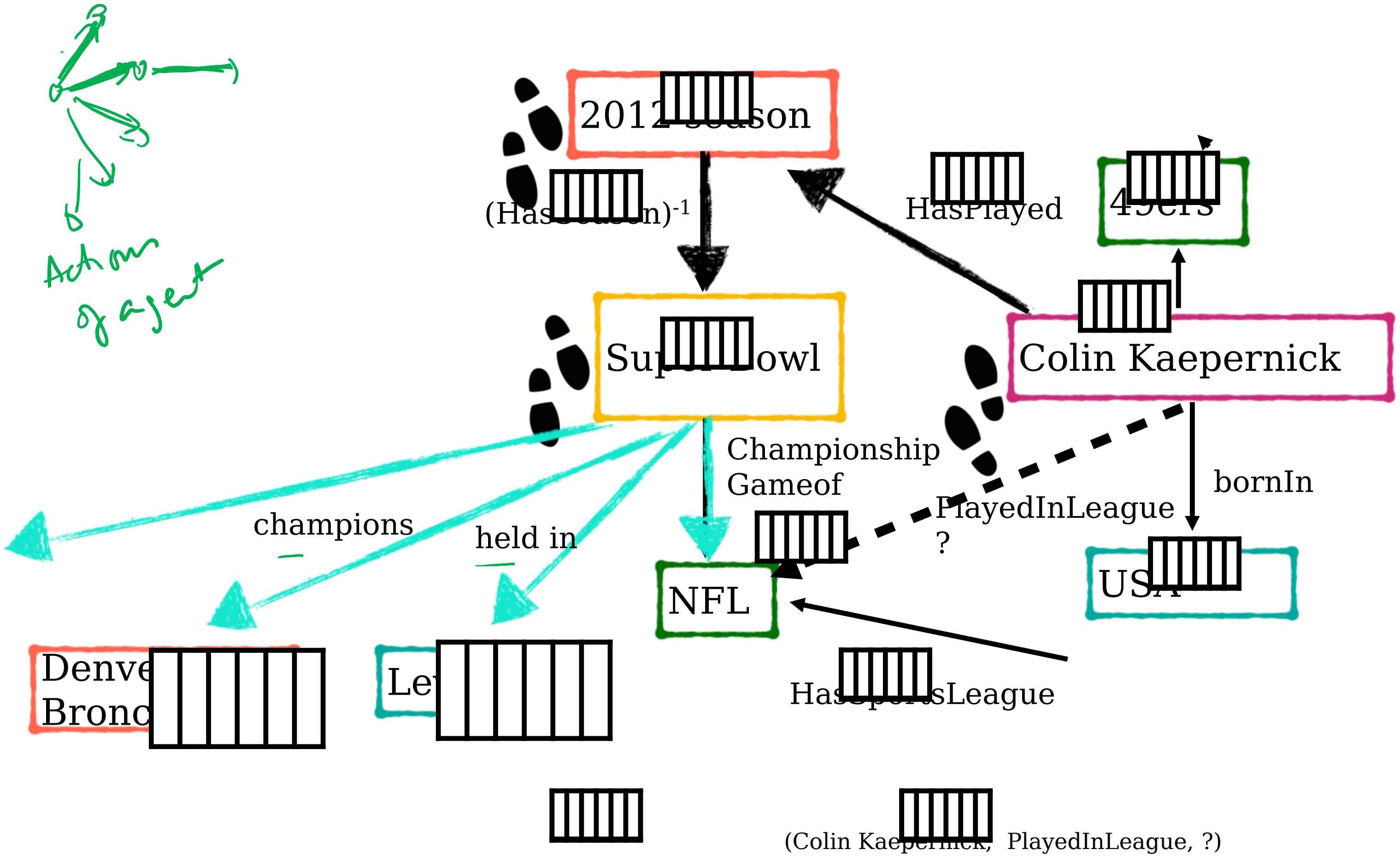
self loop

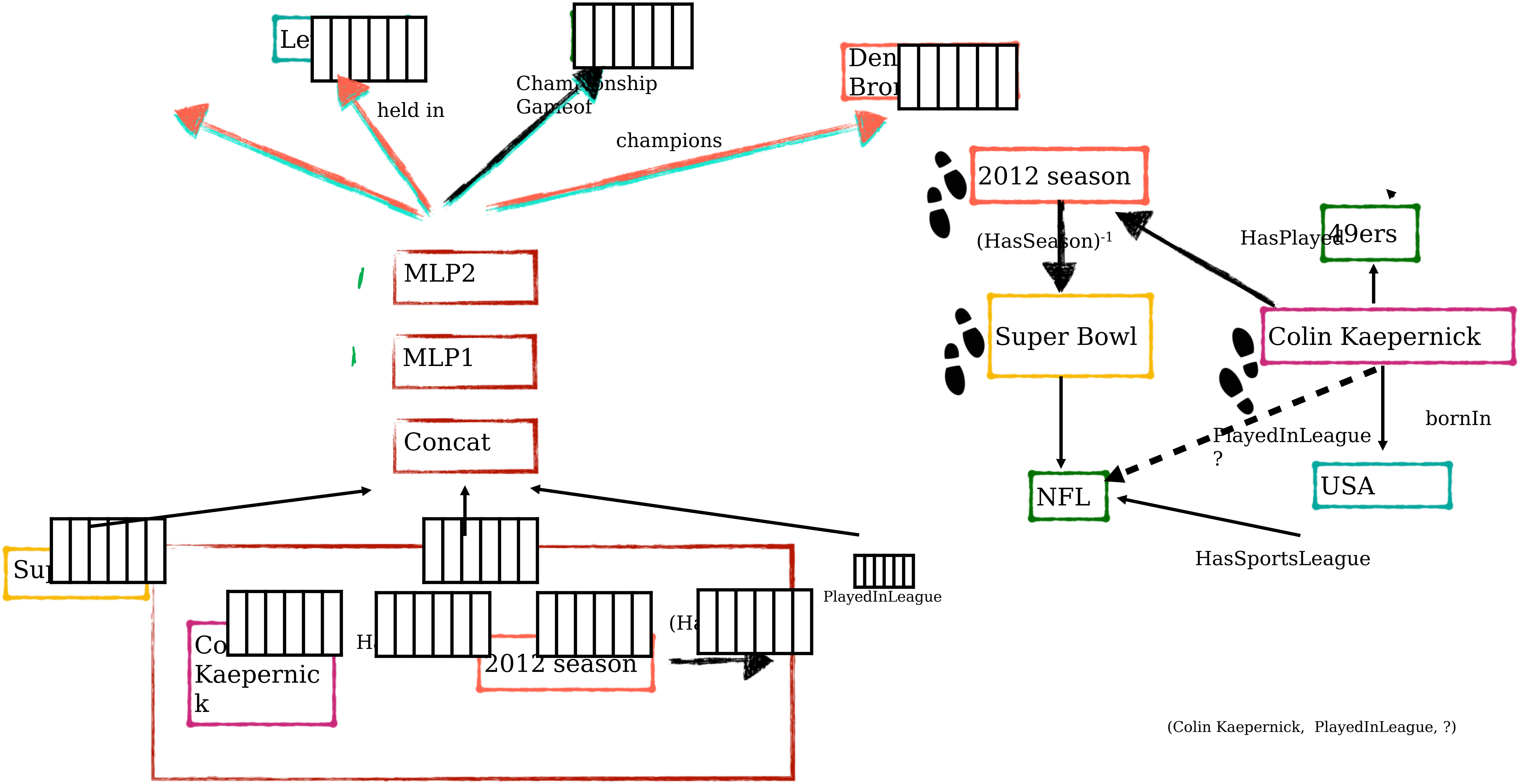


inverse edge

! father

father : inv:



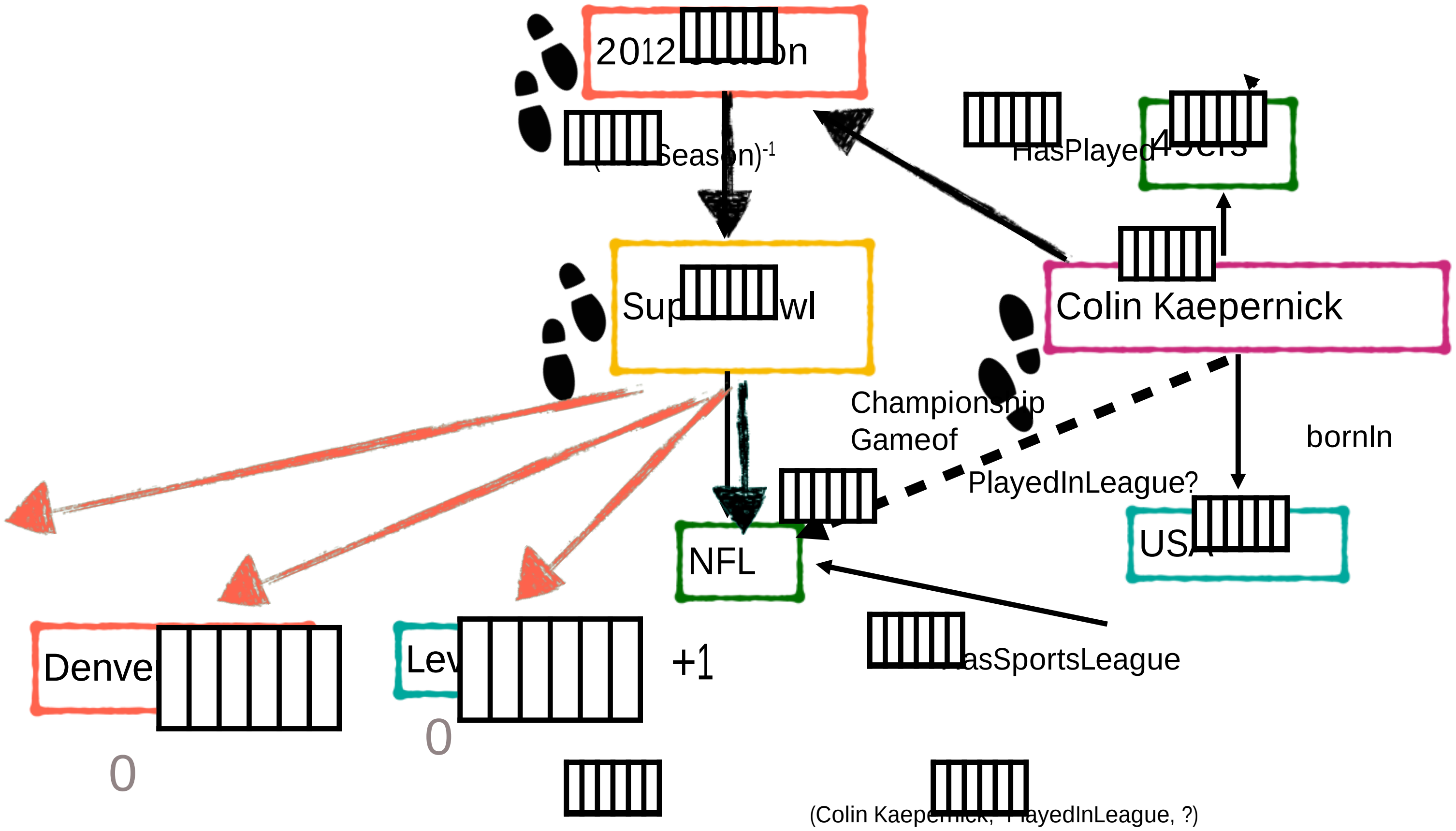


(Colin Kaepernick, PlayedInLeague, ?)

MINERVA: Rewards

- R = {+1 if we reach the answer
  0, otherwise}

noisy in reality



MINERVA: Training

Cost / loss fn. → parameter policy — 2 expectation

$$J(\theta) = \mathbb{E}_{(e_1,r,e_2)\sim D}\mathbb{E}_{A_1,\ldots,A_{T-1}\sim\pi_\theta}[R(S_T)|S_1 = (e_1, e_1, r, e_2)]$$

$\to e_t$

- Trained using Policy Gradients

$$\nabla_\theta \mathrm{J}\left(\theta\right) \sim \sum_s \mu\left(s\right) \sum_a q_\pi\left(s,a\right) \nabla_\theta \pi\left(a \mid s,\theta\right)$$

(Sutton, McAllester, Singh, Mansour, 2000)

- Monte Carlo Policy Gradients

$$\theta \leftarrow \theta + \alpha \mathrm{G}_t \nabla_\theta \log \pi\left(A_t \mid S_t, \theta\right)$$

(Williams, 1992)

- Monte Carlo Policy Gradients with Control Variates

$$\theta \leftarrow \theta + \alpha\left(\mathrm{G}_t - b\left(S_t\right)\right) \nabla_\theta \log \pi\left(A_t \mid S_t, \theta\right)$$

REINFORCE with baseline

AQA: $\mathbb{E}\ldots\left[R(\overset{\to a}{f(q)}) - \overset{\text{baseline}}{B(q_0)}\right] + \lambda H\left[\pi(q \mid q_0)\right]$

- Entropy regularization to sample more diverse paths.

Programming assignment

- Answer questions over toy KG and corpora
- Free to use any tools and resources
- Video to demonstrate method and answering outputs
- 10 dev and 10 test questions
- Comment on strengths and weaknesses from error analysis on test set
- Deadline: 21 July 2020, 14:00

Fun

simple

open-ended

word2vec, POS, DP, ..., NERD, prog. language

→ Zoom free

5-minute max

unsupervised, tuning

no training

Hackathon

send me the YouTube / Google Drive / OneDrive / Dropbox link

Conclusions

- Reinforcement learning for QA has a high potential
- Many possibilities, not explored well yet

  2017 / 2018

- Needs modeling: Agent, environment, rewards, states, policy, value

AQA

- Coercing answers from reformulated questions is a viable strategy for open-domain QA

MINERVA

- RL has showed success in KG reasoning