PADERBORN UNIVERSITY

Answering Causal Questions With Reinforcement Learning

Lukas Blübaum

Supervisor: Dr. Stefan Heindorf

DICE

Data Science Group
Paderborn University

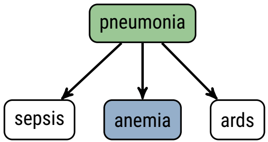
Causal Question Answering

- Causal Questions
  - Determine relationships between causes and effects
  - E.g. understand what effects a cause could have in the future

Binary Causal Questions
Does pneumonia cause anemia?

Open-Ended Questions
What can cause anemia?

Comprehension Questions
How does pneumonia cause anemia?



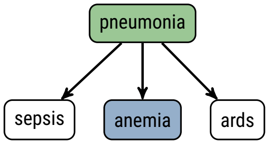
Causal Question Answering

- Causal Questions
  - Determine relationships between causes and effects
  - E.g. understand what effects a cause could have in the future
- Use Cases
  - Search engines
  - Virtual assistants like Alexa
  - Automated decision-making

Binary Causal Questions
Does pneumonia cause anemia?

Open-Ended Questions
What can cause anemia?

Comprehension Questions
How does pneumonia cause anemia?



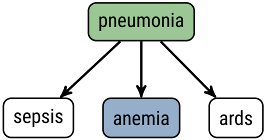
Causal Question Answering

- Causal Questions
  - Determine relationships between causes and effects
  - E.g. understand what effects a cause could have in the future
- Use Cases
  - Search engines
  - Virtual assistants like Alexa
  - Automated decision-making
- Limitations of prior approaches
  - Often not explainable
  - Lack of large-scale datasets of causal relations of high quality

Binary Causal Questions
Does pneumonia cause anemia?

Open-Ended Questions
What can cause anemia?

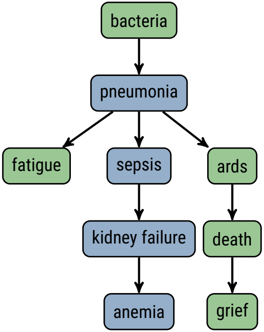
Comprehension Questions
How does pneumonia cause anemia?



Causal Question Answering

- Solution
  - Perform walks on a knowledge graph
  - Formulated as a sequential decision problem via reinforcement learning

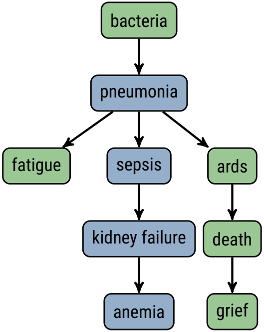
Does pneumonia cause anemia?



Causal Question Answering

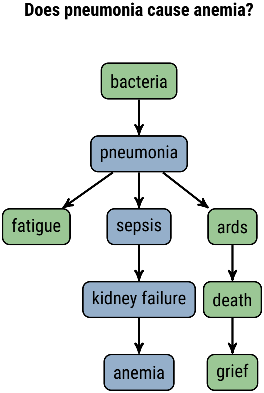
- Solution
  - Perform walks on a knowledge graph
  - Formulated as a sequential decision problem via reinforcement learning
- Advantages
  - Answers are explainable
    $\Rightarrow$ Can follow the reasoning chain
  - Can answer binary and open-ended questions

Does pneumonia cause anemia?



Causal Question Answering

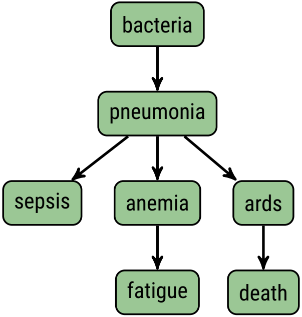
- Solution
  - Perform walks on a knowledge graph
  - Formulated as a sequential decision problem via reinforcement learning
- Advantages
  - Answers are explainable
    $\Rightarrow$ Can follow the reasoning chain
  - Can answer binary and open-ended questions
- Contributions
  - Approach to answer binary causal questions via reinforcement learning
  - Introduce an Actor-Critic (A2C) based agent with generalized advantage estimation (GAE)
  - Supervised learning and reward shaping to deal with large action spaces and sparse rewards



CauseNet

Large-Scale Causal Knowledge Graph

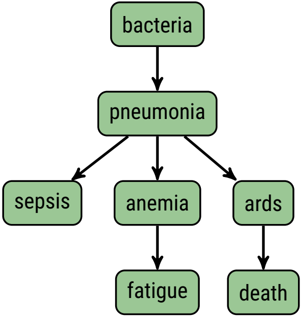
- $\mathcal{K} = (\mathcal{E}, \mathcal{R})$: entities $\mathcal{E}$, relations $\mathcal{R}$
- $\mathcal{R} = \{mayCause\}$:
  $\Rightarrow$ Claimed causal relations
- Meta-information like the original sentence and the URL for each relation
- Two configurations: CauseNet-Precision and CauseNet-Full [Heindorf et al., 2020]



CauseNet

Large-Scale Causal Knowledge Graph

- $\mathcal{K} = (\mathcal{E}, \mathcal{R})$: entities $\mathcal{E}$, relations $\mathcal{R}$
- $\mathcal{R} = \{mayCause\}$:
  $\Rightarrow$ Claimed causal relations
- Meta-information like the original sentence and the URL for each relation
- Two configurations: CauseNet-Precision and CauseNet-Full [Heindorf et al., 2020]



Causal Questions

- Does pneumonia cause fatigue?
- Is sepsis caused by pneumonia?

Task and Problem Definition

Causal Question Answering on Knowledge Graph $\mathcal{K}$

- Input:
  - $\mathcal{K} = (\mathcal{E}, \mathcal{R})$ with $\mathcal{R} = \{cause\}$
  - Question q with exactly one cause e_c and one effect e_e
    Does pneumonia (= e_c) cause anemia (= e_e)?

Task and Problem Definition

Causal Question Answering on Knowledge Graph $\mathcal{K}$

- Input:
  - $\mathcal{K} = (\mathcal{E}, \mathcal{R})$ with $\mathcal{R} = \{cause\}$
  - Question q with exactly one cause e_c and one effect e_e
    Does pneumonia ($= e_c$) cause anemia ($= e_e$)?
- Output:
  - Path: $(e_c, e_1, \ldots, e_e)$ with $e_c, e_i, e_e \in \mathcal{E}$
  - If such a path can be found answer “yes” and “no” otherwise

Task and Problem Definition

Causal Question Answering on Knowledge Graph $\mathcal{K}$

- Input:
  - $\mathcal{K} = (\mathcal{E}, \mathcal{R})$ with $\mathcal{R} = \{cause\}$
  - Question q with exactly one cause e_c and one effect e_e
    Does pneumonia (= e_c) cause anemia (= e_e)?
- Output:
  - Path: $(e_c, e_1, \ldots, e_e)$ with $e_c, e_i, e_e \in \mathcal{E}$
  - If such a path can be found answer "yes" and "no" otherwise
- Challenges:
  - Only one relation type contrary to prior approaches
    $\Rightarrow$ Large action space
  - CauseNet-Precision has 80,223 entities

Reinforcement Learning

- Policy gradient methods with policy network $\pi_\theta(a_t|s_t)$:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log(\pi_\theta(a_t|s_t))\Psi_t\right]$$

Reinforcement Learning

- Policy gradient methods with policy network $\pi_\theta(a_t|s_t)$:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log(\pi_\theta(a_t|s_t))\Psi_t\right]$$

- Possible forms for Ψ_t [Schulman et al., 2016]:

$$R_t = \sum_{i=0}^{T-t} \gamma^i r_{t+i} \qquad \mathcal{A}_t^{\psi} = R_t(\lambda) - V_\psi(s_t)$$

Reinforcement Learning

- Policy gradient methods with policy network $\pi_\theta(a_t|s_t)$:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log(\pi_\theta(a_t|s_t))\Psi_t\right]$$

- Possible forms for Ψ_t [Schulman et al., 2016]:

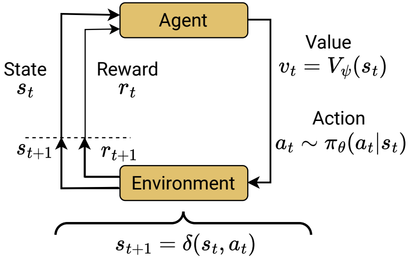
$$R_t = \sum_{i=0}^{T-t} \gamma^i r_{t+i} \qquad \mathcal{A}_t^\psi = R_t(\lambda) - V_\psi(s_t)$$

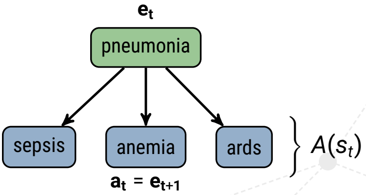
- Monte-Carlo return R_t unbiased but high variance
  $\Rightarrow$ REINFORCE
- Advantage $\mathcal{A}_t^\psi$ introduces value network $V_\psi(s_t)$ to reduce variance
  $\Rightarrow$ Advantage Actor-Critic (A2C) [Mnih et al., 2016]
- Using the λ-return $R_t(\lambda)$ in the advantage yields the generalized advantage estimation (GAE) [Schulman et al., 2016]

Environment

Agent, States, Actions

- Agent
  - $\pi_\theta(a_t|s_t)$: policy network
    $\Rightarrow$ Distribution over actions
  - $V_\psi(s_t)$: value network
    $\Rightarrow$ Reward from state s_t onwards

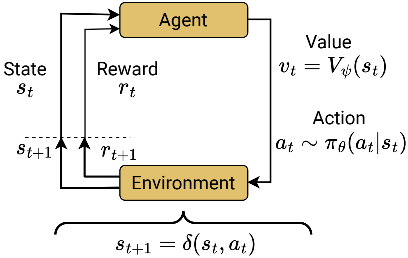


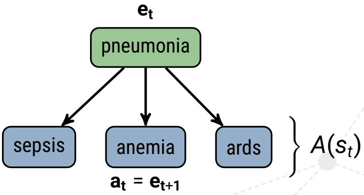


Environment

Agent, States, Actions

- Agent
  - $\pi_\theta(a_t|s_t)$: policy network
    $\Rightarrow$ Distribution over actions
  - $V_\psi(s_t)$: value network
    $\Rightarrow$ Reward from state s_t onwards
- States
  - $s = (\mathbf{q}, e_t, \mathbf{e_t}, \mathbf{h_t}, e_e)$
  - Question Embedding **q**
  - Current entity e_t, target entity e_e
  - History $\mathbf{h_t}$ (LSTM hidden states)

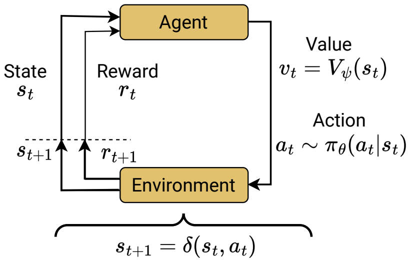


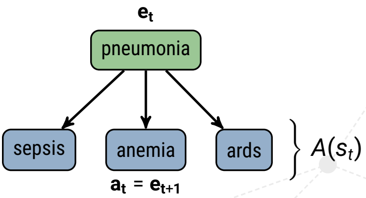


Environment

Agent, States, Actions

- Agent
  - $\pi_\theta(a_t|s_t)$: policy network
    $\Rightarrow$ Distribution over actions
  - $V_\psi(s_t)$: value network
    $\Rightarrow$ Reward from state s_t onwards
- States
  - $s = (\mathbf{q}, e_t, \mathbf{e_t}, \mathbf{h_t}, e_e)$
  - Question Embedding **q**
  - Current entity e_t, target entity e_e
  - History $\mathbf{h_t}$ (LSTM hidden states)
- Actions
  - All neighboring entities of e_t:
    $A(s_t) = \{e|(e_t, r, e) \in \mathcal{K}\}$

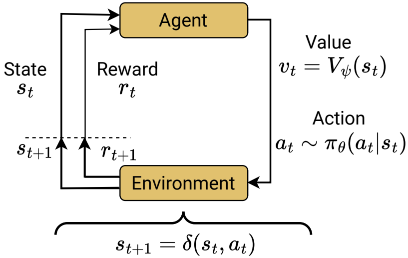


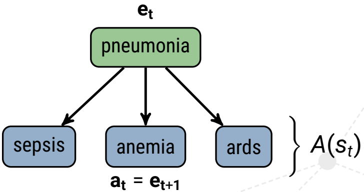


Environment

Actions, Transitions, Rewards

- Actions
  - *STAY* action:
    $A(s_t) = A(s_t) \cup \{STAY\}$
  - Including inverse edges

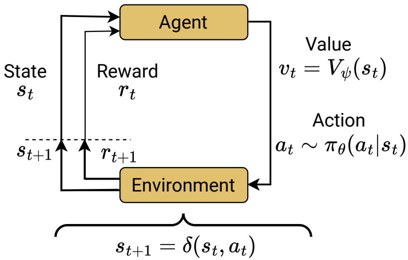


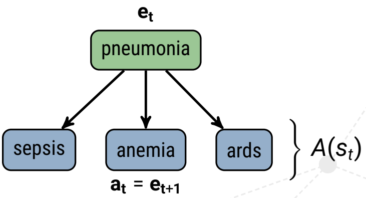


Environment

Actions, Transitions, Rewards

- Actions
  - *STAY* action: $A(s_t) = A(s_t) \cup \{STAY\}$
  - Including inverse edges
- Transitions
  - $\delta(s_t, a_t) = s_{t+1}$ where $a_t = e_{t+1}$
  - Deterministic and entirely defined by the graph

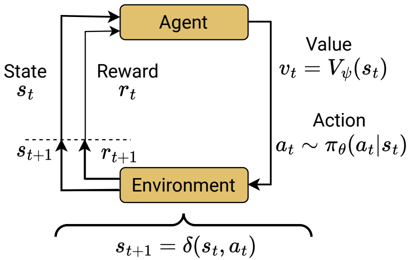


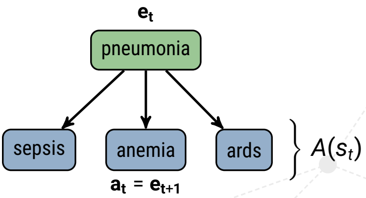


Environment

Actions, Transitions, Rewards

- Actions
  - *STAY* action:
    $A(s_t) = A(s_t) \cup \{STAY\}$
  - Including inverse edges
- Transitions
  - $\delta(s_t, a_t) = s_{t+1}$ where $a_t = e_{t+1}$
  - Deterministic and entirely defined by the graph
- Rewards
  - $\mathcal{R}(s_{T-1}) = r_t$ with $r_t = 1$ if $e_{T-1} = e_e$ and $r_t = 0$ otherwise
  - 0 at all other time steps
    $\Rightarrow$ sparse reward





Agent

Training

- Network Architecture
  - LSTM to include the path history
  - Stack two feedforward networks on top
    $\Rightarrow$ For policy network and value network

Agent

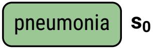
Training

- Network Architecture
  - LSTM to include the path history
  - Stack two feedforward networks on top
    $\Rightarrow$ For policy network and value network
- Preprocessing of questions
  - No negative information in CauseNet
    $\Rightarrow$ Only use positive questions
  - Find e_c and e_e in graph via exact string matching

Agent

Training

- Network Architecture
  - LSTM to include the path history
  - Stack two feedforward networks on top
    $\Rightarrow$ For policy network and value network
- Preprocessing of questions
  - No negative information in CauseNet
    $\Rightarrow$ Only use positive questions
  - Find e_c and e_e in graph via exact string matching
- Sample rollouts
  - Starting at e_c sample rollouts of length T
  - If e_e found, the agent should use the *STAY* action

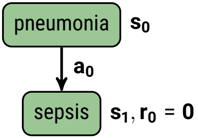


rollout = (

Agent

Training

- Network Architecture
  - LSTM to include the path history
  - Stack two feedforward networks on top
    $\Rightarrow$ For policy network and value network
- Preprocessing of questions
  - No negative information in CauseNet
    $\Rightarrow$ Only use positive questions
  - Find e_c and e_e in graph via exact string matching
- Sample rollouts
  - Starting at e_c sample rollouts of length T
  - If e_e found, the agent should use the *STAY* action

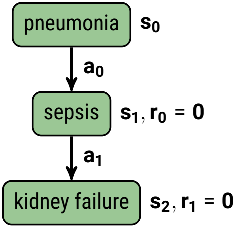


$$\text{rollout} = ((s_0, a_0, r_0),$$

Agent

Training

- Network Architecture
  - LSTM to include the path history
  - Stack two feedforward networks on top
    $\Rightarrow$ For policy network and value network
- Preprocessing of questions
  - No negative information in CauseNet
    $\Rightarrow$ Only use positive questions
  - Find e_c and e_e in graph via exact string matching
- Sample rollouts
  - Starting at e_c sample rollouts of length T
  - If e_e found, the agent should use the *STAY* action

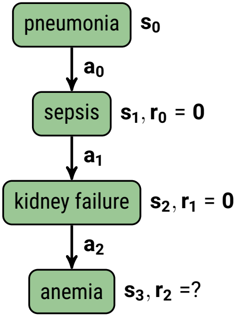


$$\text{rollout} = ((s_0, a_0, r_0), (s_1, a_1, r_1),$$

Agent

Training

- Network Architecture
  - LSTM to include the path history
  - Stack two feedforward networks on top
    $\Rightarrow$ For policy network and value network
- Preprocessing of questions
  - No negative information in CauseNet
    $\Rightarrow$ Only use positive questions
  - Find e_c and e_e in graph via exact string matching
- Sample rollouts
  - Starting at e_c sample rollouts of length T
  - If e_e found, the agent should use the *STAY* action



$$\text{rollout} = ((s_0, a_0, r_0), (s_1, a_1, r_1), (s_2, a_2, r_2))$$

Agent

Update Rules

- Using Synchronous Advantage Actor-Critic (A2C) with GAE
- Actor: policy network $\pi_\theta(a_t|s_t)$:

$$\nabla_\theta J(\theta) = -\frac{1}{B}\sum_{i}^{B}\sum_{t=0}^{T-2}\nabla_\theta \log(\pi_\theta(a_t|s_t))\,\mathcal{A}_t^{\psi} + \beta H_{\pi_\theta}$$

  - B: batch size, T: episode length, $\mathcal{A}_t^{\psi}$: GAE
  - H_{π_θ}: entropy regularization $\Rightarrow$ exploration vs. exploitation

Agent

Update Rules

- Using Synchronous Advantage Actor-Critic (A2C) with GAE
- Actor: policy network $\pi_\theta(a_t|s_t)$:

$$\nabla_\theta J(\theta) = -\frac{1}{B}\sum_{i}^{B}\sum_{t=0}^{T-2}\nabla_\theta \log(\pi_\theta(a_t|s_t))\,\mathcal{A}_t^{\psi} + \beta H_{\pi_\theta}$$

  - B: batch size, T: episode length, $\mathcal{A}_t^{\psi}$: GAE
  - H_{π_θ}: entropy regularization $\Rightarrow$ exploration vs. exploitation
- Critic: value network $V_\psi(s_t)$:

$$\nabla_\psi J(\psi) = \frac{1}{B(T-1)}\sum_{i}^{B}\sum_{t=0}^{T-2}\nabla_\psi (R_t(\lambda) - V_\psi(s_t))^2$$

  - MSE between λ-return and value network predictions $V_\psi(s_t)$

Agent

Inference

Inference Time

- Receive positive and negative questions
- Answer “yes” if a path was found and “no” otherwise

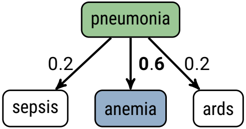
Agent

Inference

Inference Time

- Receive positive and negative questions
- Answer “yes” if a path was found and “no” otherwise

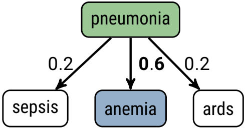
- Greedy decoding:
  - Choose action: $arg\,max_{a_t \in \mathcal{A}(s_t)} \pi_\theta(a_t|s_t)$
  - Myopic behavior



Inference Time

- Receive positive and negative questions
- Answer “yes” if a path was found and “no” otherwise

- Greedy decoding:
  - Choose action: $arg\,max_{a_t \in \mathcal{A}(s_t)} \pi_\theta(a_t|s_t)$
  - Myopic behavior



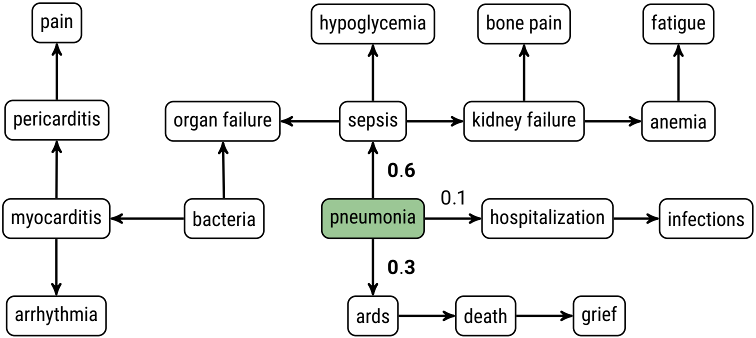
- Beam search:
  - Always keep set of best partial solutions (paths)
  - Paths ranked by their probability
  - Probability of path $p = (e_c, e_1, \ldots, e_{T-1})$ with $e_t = a_{t-1}$:

$$\mathbb{P}(p) = \prod_{t=0}^{T-2} \pi_\theta(a_t|s_t)$$

Walkthrough

Beam Width = 2

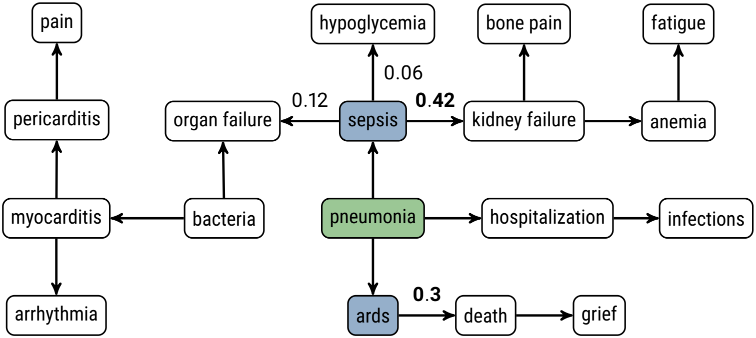
Question: Does pneumonia cause anemia?



Walkthrough

Beam Width = 2

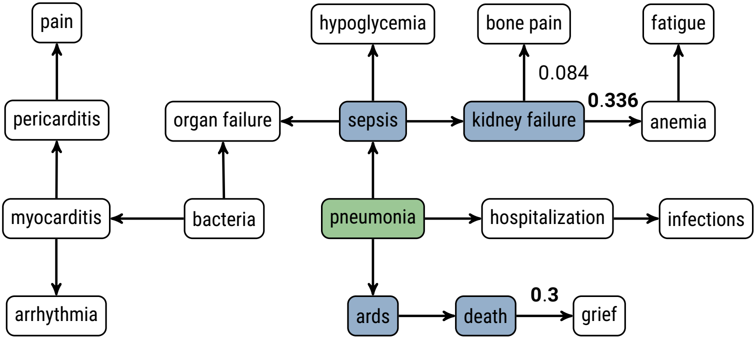
Question: Does pneumonia cause anemia?



Walkthrough

Beam Width = 2

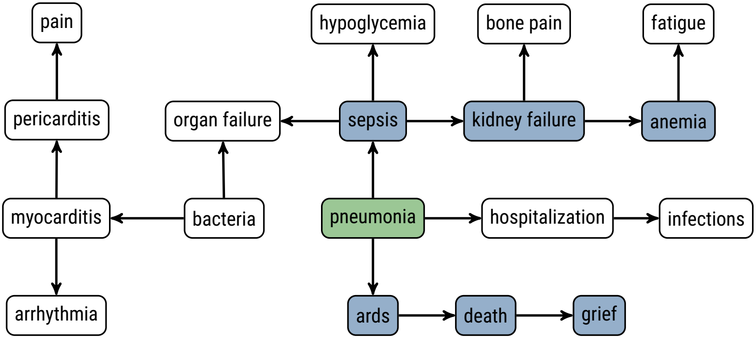
Question: Does pneumonia cause anemia?



Walkthrough

Beam Width = 2

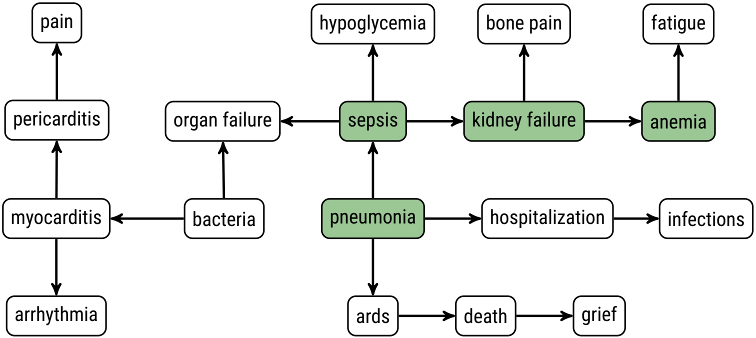
Question: Does pneumonia cause anemia?



Walkthrough

Beam Width = 2

Question: Does pneumonia cause anemia?

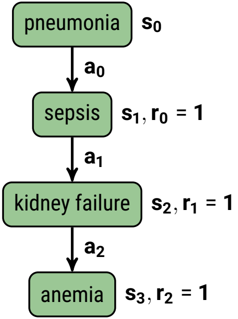


Bootstrapping with Supervised Learning

- Problem: large action space + sparse rewards
  - Slow convergence
  - Guidance in the beginning

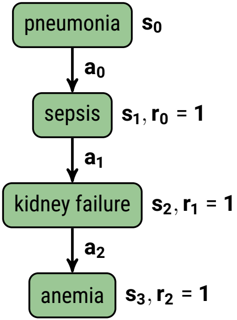
Bootstrapping with Supervised Learning

- Problem: large action space + sparse rewards
  - Slow convergence
  - Guidance in the beginning
- Generating expert demonstrations [Xiong et al., 2017]:
  - Paths from a breadth-first search (BFS)
  - Preprocessing step for each question q
    Find path between e_c and e_e



Bootstrapping with Supervised Learning

- Problem: large action space + sparse rewards
  - Slow convergence
  - Guidance in the beginning
- Generating expert demonstrations [Xiong et al., 2017]:
  - Paths from a breadth-first search (BFS)
  - Preprocessing step for each question q
    Find path between e_c and e_e
- Supervised gradient update:
  - $r_t = 1$ at each time step
  - Batch size B, episode length T
  - Entropy regularization H_{π_θ}



$$\nabla_\theta J(\theta) = -\frac{1}{B} \sum_{i}^{B} \sum_{t=0}^{T-2} \nabla_\theta \log(\pi_\theta(a_t|s_t))\, r_t + \beta H_{\pi_\theta}$$

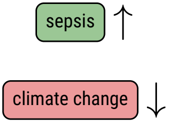
Reward Shaping

- Problem: sparse rewards
  - Agent only receives a reward if $e_{T-1} = e_e$
  - Rarely finds correct paths at the start

Reward Shaping

- Problem: sparse rewards
  - Agent only receives a reward if $e_{T-1} = e_e$
  - Rarely finds correct paths at the start
- Introduce auxiliary reward:
  - Score last node according to its relevance to the question [Yasunaga et al., 2021]

Does pneumonia cause anemia?



Reward Shaping

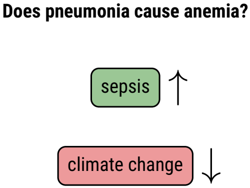
- Problem: sparse rewards
  - Agent only receives a reward if $e_{T-1} = e_e$
  - Rarely finds correct paths at the start
- Introduce auxiliary reward:
  - Score last node according to its relevance to the question [Yasunaga et al., 2021]
- Updated reward:

$$\mathcal{R}'(s_{T-1}) = \begin{cases} \mathcal{R}(s_{T-1}), & \mathcal{R}(s_{T-1}) = 1 \\ Score(q, e_{T-1}) \cdot \omega, & \text{otherwise} \end{cases}$$

- Where *Score* is defined as:

$$Score(q, e_{T-1}) = LM_{head}(LM_{enc}([q; e_{T-1}]))$$

  - LM_{head}: feedforward network, LM_{enc}: language model encoder



Evaluation

MS MARCO					
	Accuracy	**F_1**	**Recall**	**Precision**	**|Nodes|**
Agent 2-Hop	0.460	0.562	0.408	**0.901**	**26**
Agent 3-Hop	0.529	0.648	0.511	0.884	27
BFS 2-Hop	0.494	0.612	0.471	0.875	1,727
BFS 3-Hop	0.589	0.714	0.605	0.871	3,339
UnifiedQA-v2	**0.722**	**0.828**	**0.789**	0.871	–

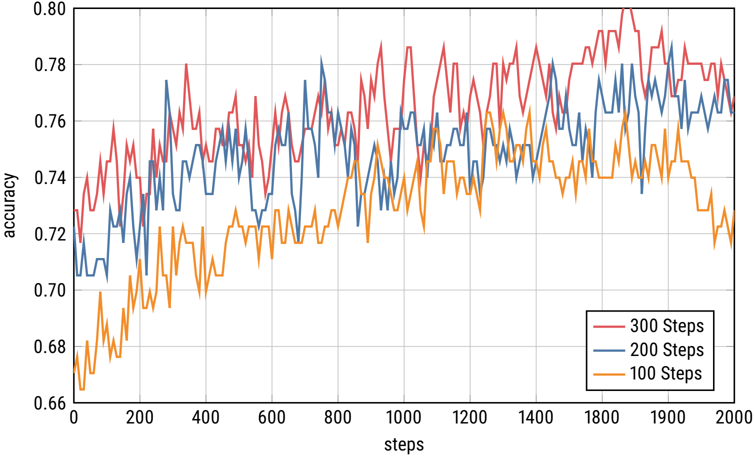
SemEval					
	Accuracy	**F_1**	**Recall**	**Precision**	**|Nodes|**
Agent 2-Hop	0.769	0.714	0.575	**0.943**	**27**
Agent 3-Hop	0.775	0.727	0.598	0.929	29
BFS 2-Hop	**0.815**	**0.787**	0.678	0.937	1,565
BFS 3-Hop	0.751	0.754	0.759	0.750	3,687
UnifiedQA-v2	0.497	0.653	**0.943**	0.500	–

Ablation Study

	MS MARCO				SemEval			
	A	F_1	R	P	A	F_1	R	P
Agent 2-Hop	**0.460**	**0.562**	**0.408**	0.901	**0.769**	**0.714**	**0.575**	0.943
— Beam Search	0.293	0.306	0.184	**0.911**	0.613	0.374	0.230	**1.000**
— Supervised Learning	0.342	0.397	0.257	0.891	0.682	0.538	0.369	**1.000**
— Actor-Critic	0.441	0.539	0.386	0.896	0.740	0.657	0.494	0.977
— Inverse Edges	0.422	0.513	0.359	0.899	0.740	0.651	0.483	**1.000**
+ Reward Shaping (0.1)	0.449	0.548	0.395	0.898	0.757	0.691	0.540	0.959
+ Reward Shaping (1.0)	0.403	0.489	0.336	0.893	**0.769**	0.706	0.552	0.980

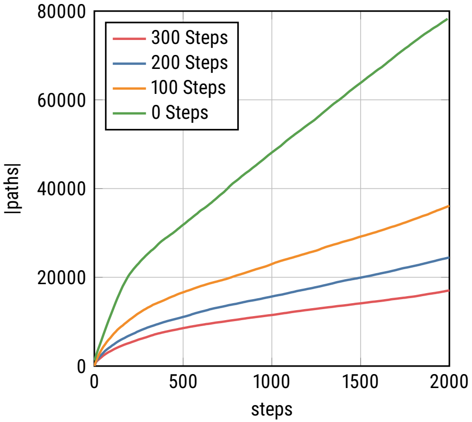
Effects of Supervised Learning

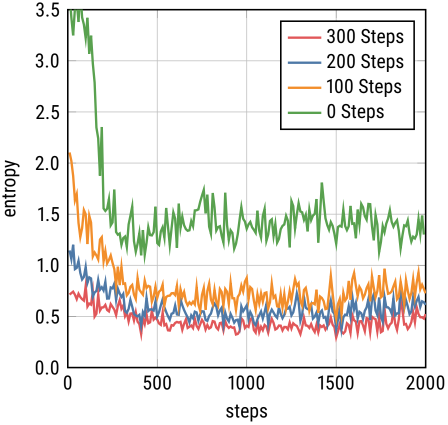
Accuracy



Effects of Supervised Learning

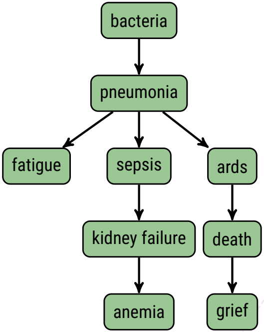
Number of Explored Paths + Entropy of Policy Network





Conclusion

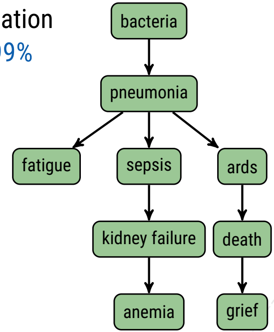
- Summary
  - Introduced an Actor-Critic (A2C) based agent to answer causal questions
  - Extended with supervised learning and reward shaping



Conclusion

- Summary
  - Introduced an Actor-Critic (A2C) based agent to answer causal questions
  - Extended with supervised learning and reward shaping
- Conclusion
  - Supervised learning provides an effective foundation
  - Effectively prunes the search space by around 99%
  - Paths can be used as explanations

    $\Rightarrow$ Use meta-information to verify the claims

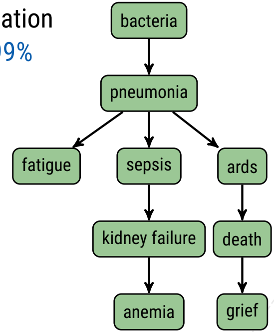


Conclusion

- Summary
  - Introduced an Actor-Critic (A2C) based agent to answer causal questions
  - Extended with supervised learning and reward shaping
- Conclusion
  - Supervised learning provides an effective foundation
  - Effectively prunes the search space by around 99%
  - Paths can be used as explanations

    $\Rightarrow$ Use meta-information to verify the claims
- Future Work
  - Extend to open-ended questions

    $\Rightarrow$Straightforward extension via majority voting
  - Add different causal knowledge graphs
  - Explore ways to consider negative causal questions during training



References I

[Heindorf et al., 2020] Heindorf, S., Scholten, Y., Wachsmuth, H., Ngomo, A.-C. N., and Potthast, M. (2020).
Causenet: Towards a causality graph extracted from the web.
In *CIKM*. ACM.

[Mnih et al., 2016] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016).
Asynchronous methods for deep reinforcement learning.
In Balcan, M. and Weinberger, K. Q., editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org.

[Schulman et al., 2016] Schulman, J., Moritz, P., Levine, S., Jordan, M. I., and Abbeel, P. (2016).
High-dimensional continuous control using generalized advantage estimation.
In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[Xiong et al., 2017] Xiong, W., Hoang, T., and Wang, W. Y. (2017).
DeepPath: A reinforcement learning method for knowledge graph reasoning.
In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, Copenhagen, Denmark. Association for Computational Linguistics.

References II

[Yasunaga et al., 2021] Yasunaga, M., Ren, H., Bosselut, A., Liang, P., and Leskovec, J. (2021). QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

λ-return & GAE

- N-step returns:

$$R_t^{(n)} = \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n V_\psi(s_{t+n})$$

- Which n is best?
- λ-return exponentially-weighted average of n-step returns:

$$R_t(\lambda) = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)} = (1-\lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} R_t^{(n)} + \lambda^{T-t-1} R_t^{(T-t)}$$

Network Architecture

- LSTM:

$$\mathbf{h_t} = \begin{cases} LSTM(\mathbf{0}; [\mathbf{q}, \mathbf{e_c}]), & \text{if } t = 0 \\ LSTM(\mathbf{h_{t-1}}, [\mathbf{q}; \mathbf{e_t}]), & \text{otherwise} \end{cases}$$

- Policy network $\pi_\theta(a_t|s_t)$:

$$\pi_\theta(a_t|s_t) = \sigma(\mathbf{A_t} \times W_2 \times ReLU(W_1 \times \mathbf{h_t}))$$
$$a_t \sim Categorical(\pi_\theta(a_t|s_t))$$

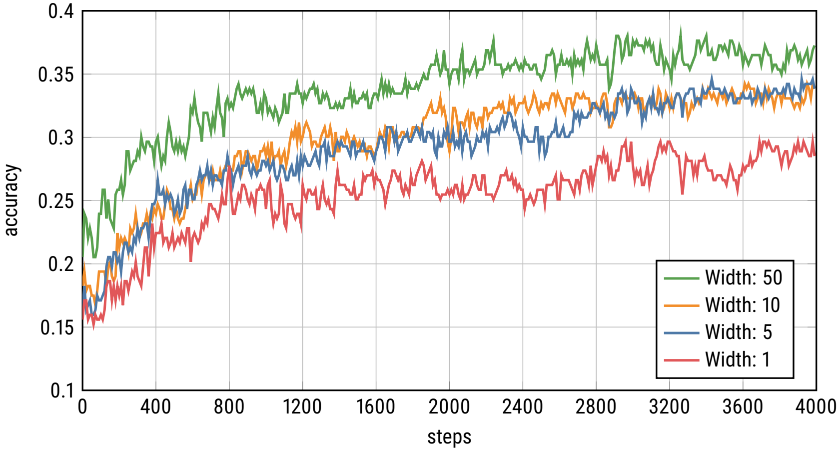
  - $\mathbf{A_t} \in \mathcal{R}^{|A(s_t)| \times 2d}$ embeddings of actions $a_t \in \mathcal{A}(s_t)$
  - σ: softmax operator
- Value network $V_\psi(s_t)$:

$$V_\psi(s_t) = W_4 \times ReLU(W_3 \times \mathbf{h_t})$$

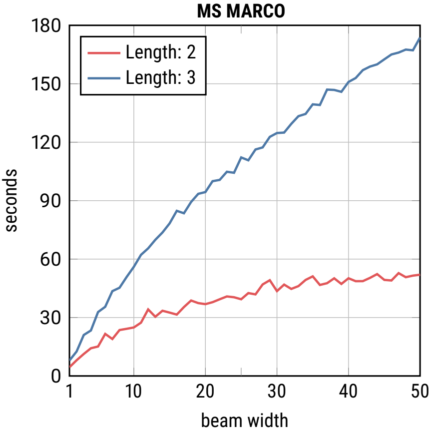
Entropy

$$H_{\pi_\theta} = \frac{1}{B(T-1)} \sum_{i}^{B} \sum_{t=0}^{T-2} \Big(- \sum_{a_t \in \mathcal{A}(s_t)} \pi_\theta(a_t|s_t) \log \pi_\theta(a_t|s_t)\Big)$$

Beam Search Evaluation

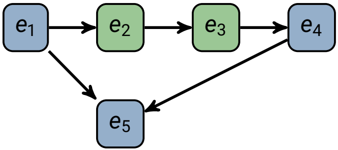


Inference Time



Inverse Edges

- Pro:
  - Undo wrong actions
  - Reach nodes that could otherwise not be reached in T steps
- Con:
  - Introduction of false positives
  - Atleast in theory it might not make sense to take an inverse edge
- Does e_1 cause e_4? and $T = 2$:



Examples

Agent 3-Hop
Cause: h. pylori **Effect:** vomiting **Path:** h. pylori $\xRightarrow{\text{cause}}$ peptic ulcer disease $\xRightarrow{\text{cause}}$ vomiting $\xRightarrow{\mathit{STAY}}$ vomiting
Cause: Xanax **Effect:** hiccups **Path:** xanax $\xRightarrow{\text{cause}}$ anxiety $\xRightarrow{\text{cause}}$ stress $\xRightarrow{\text{cause}}$ hiccups
Cause: chocolate **Effect:** constipation **Path:** chocolate $\xRightarrow{\text{cause}}$ constipation $\xRightarrow{\text{cause}}$ depression $\xRightarrow{\text{cause}^{-1}}$ constipation
Cause: rainfall **Effect:** flooding **Path:** rainfall $\xRightarrow{\text{cause}}$ flooding $\xRightarrow{\text{cause}}$ landslides $\xRightarrow{\mathit{STAY}}$ landslides