Offline RL for Natural Language Generation with Implicit Language Q Learning

Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine

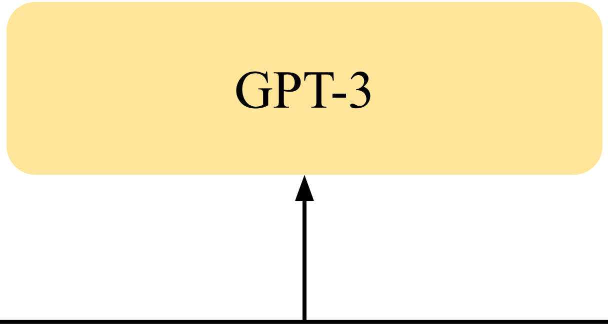
UC Berkeley

Large Language Models

GPT-3

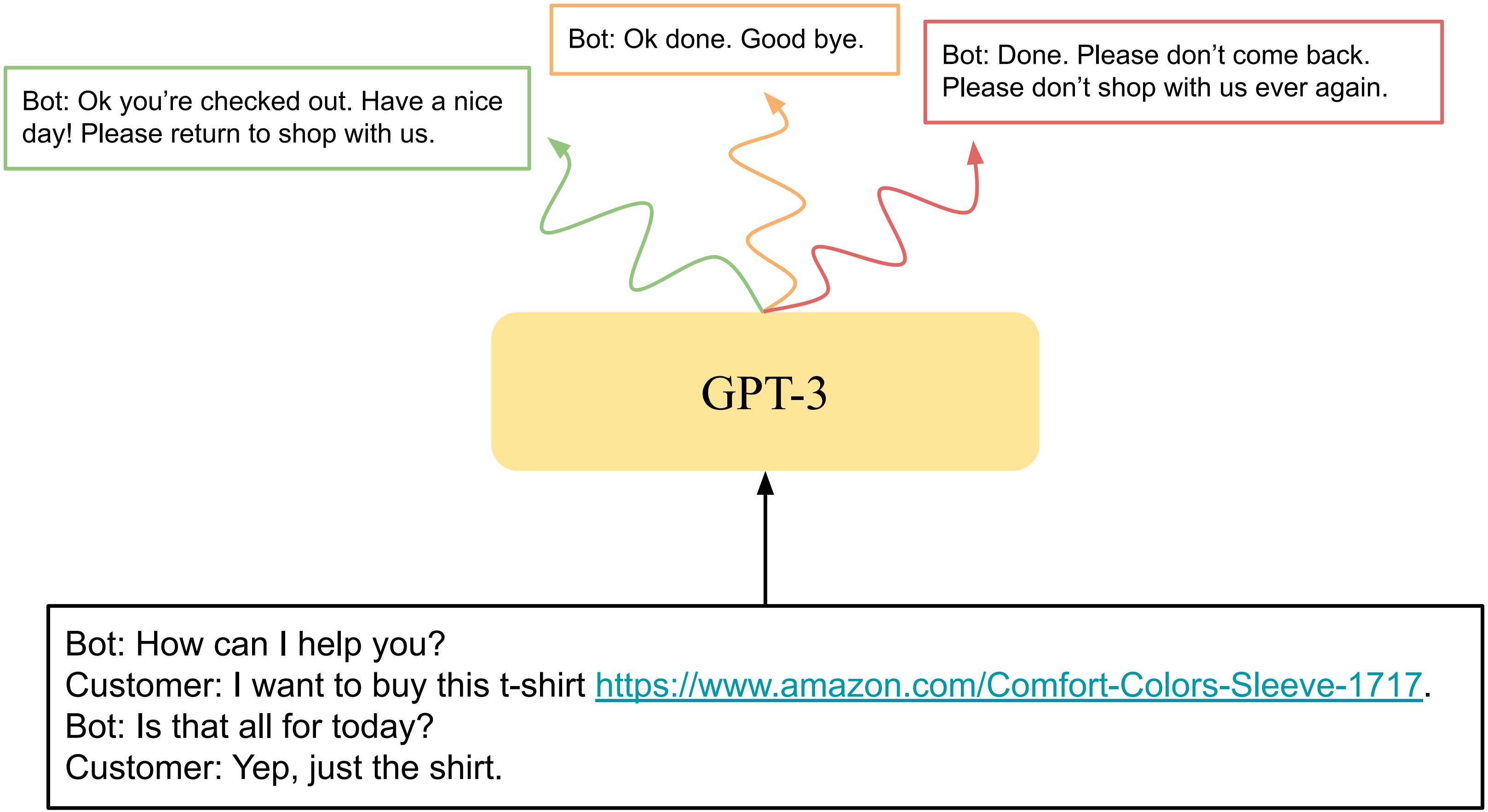
Bot: How can I help you?
Customer: I want to buy this t-shirt https://www.amazon.com/Comfort-Colors-Sleeve-1717.
Bot: Is that all for today?
Customer: Yep, just the shirt.

Large Language Models



Bot: How can I help you?
Customer: I want to buy this t-shirt https://www.amazon.com/Comfort-Colors-Sleeve-1717.
Bot: Is that all for today?
Customer: Yep, just the shirt.

Large Language Models



Large Language Models

GPT-3

Prompt: You are a helpful, kind, and efficient customer service bot.

Bot: How can I help you?
Customer: I want to buy this t-shirt https://www.amazon.com/Comfort-Colors-Sleeve-1717.
Bot: Is that all for today?
Customer: Yep, just the shirt.

Large Language Models

GPT-3

Prompt: You are a helpful, kind, and efficient customer service bot.

Bot: How can I help you?
Customer: I want to buy this t-shirt https://www.amazon.com/Comfort-Colors-Sleeve-1717.
Bot: Is that all for today?
Customer: Yep, just the shirt.

Large Language Models

Filtered Fine Tuning: (%BC)

1. Create a dataset of purely positive interactions.
2. Finetune on the data.

Good bye.

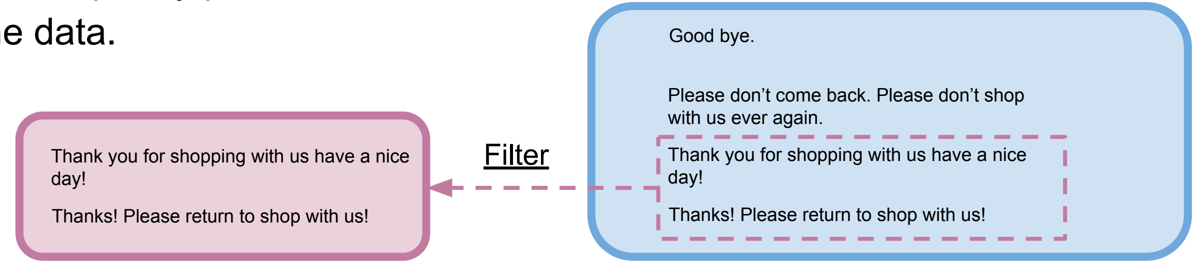
Please don't come back. Please don't shop with us ever again.

Thank you for shopping with us have a nice day!

Thanks! Please return to shop with us!

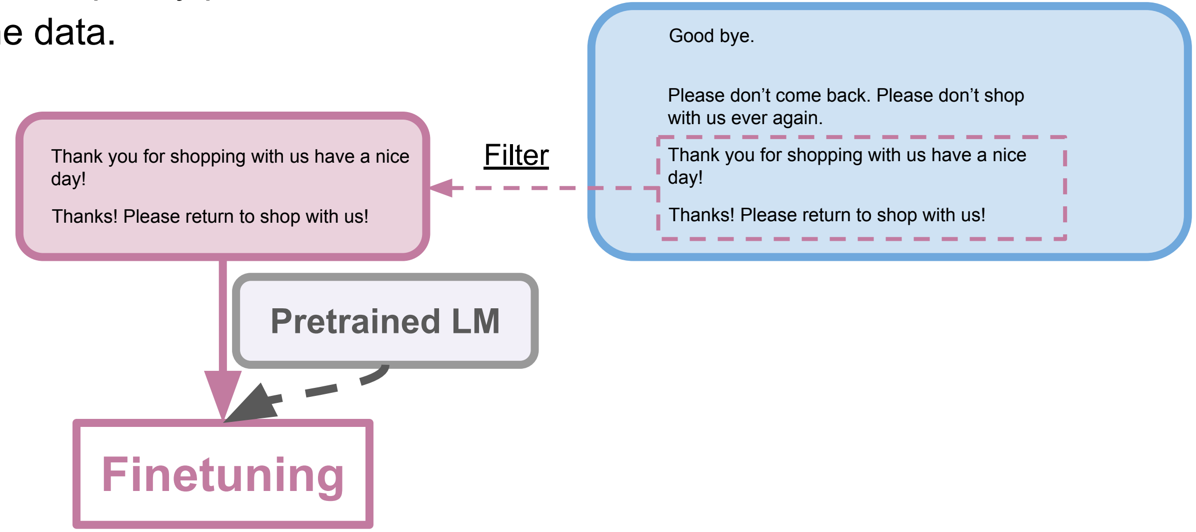
Filtered Fine Tuning: (%BC)

1. Create a dataset of purely positive interactions.
2. Finetune on the data.



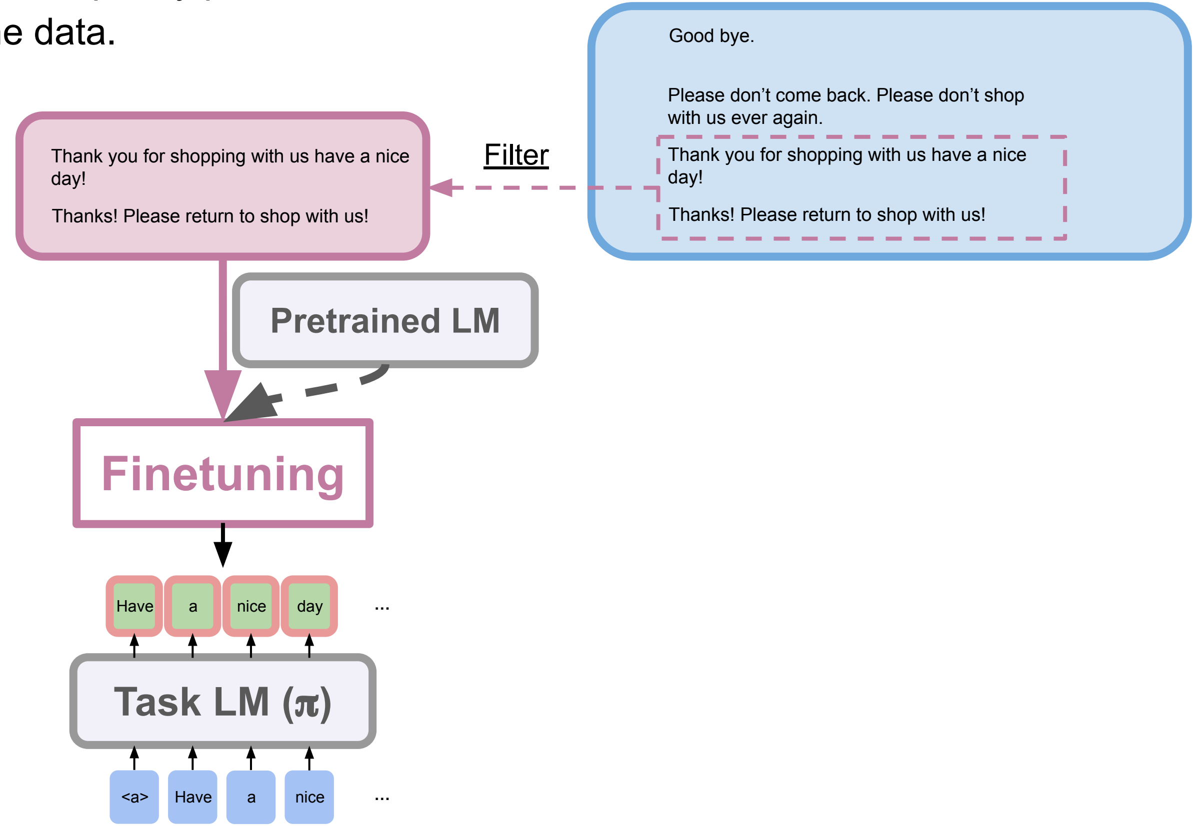
Filtered Fine Tuning: (%BC)

1. Create a dataset of purely positive interactions.
2. Finetune on the data.



Filtered Fine Tuning: (%BC)

1. Create a dataset of purely positive interactions.
2. Finetune on the data.



Offline RL: a new paradigm for fine tuning LMs

Why not directly optimize for positive interactions instead?

Suboptimal Dataset with Rewards

Good bye.

Please don’t come back. Please don’t shop with us ever again.

Thank you for shopping with us have a nice day!

Thanks! Please return to shop with us!

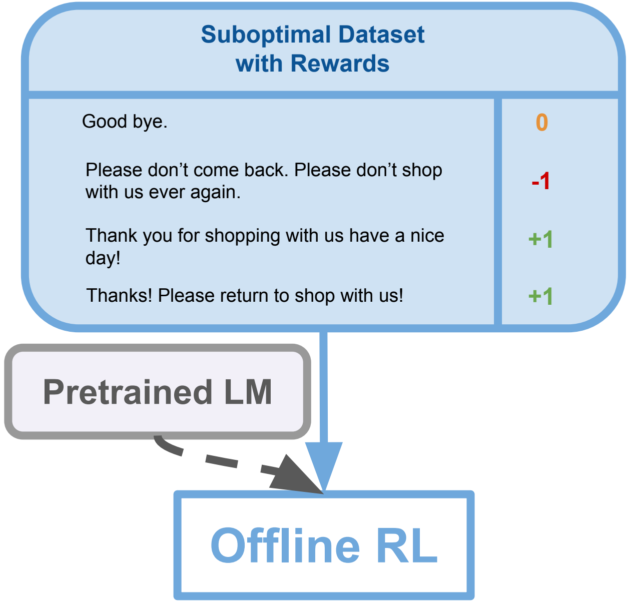
Offline RL: a new paradigm for fine tuning LMs

Suboptimal Dataset with Rewards	
Good bye.	0
Please don’t come back. Please don’t shop with us ever again.	-1
Thank you for shopping with us have a nice day!	+1
Thanks! Please return to shop with us!	+1

Why not directly optimize for positive interactions instead?

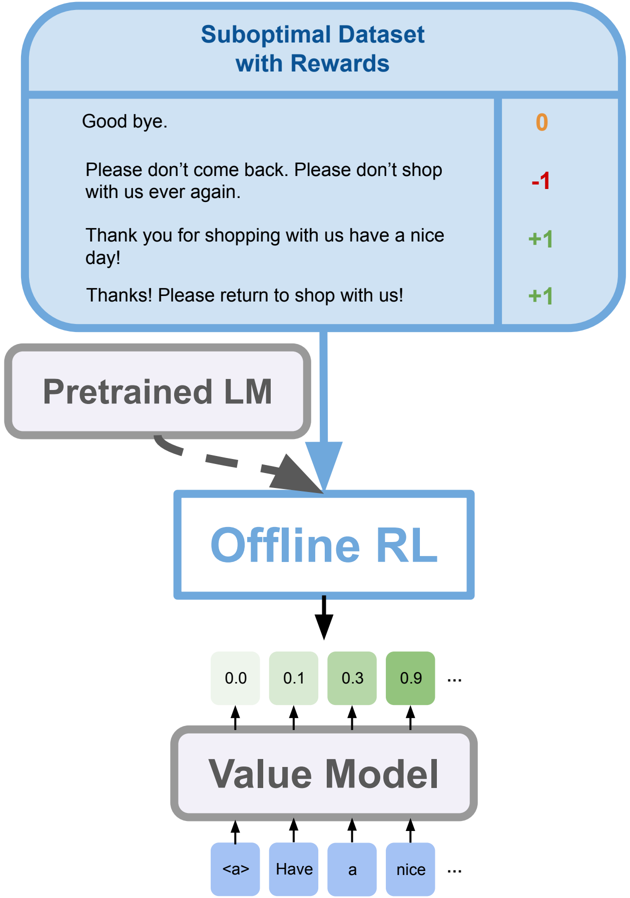
Offline RL: a new paradigm for fine tuning LMs

Why not directly optimize for positive interactions instead?



Offline RL: a new paradigm for fine tuning LMs

Why not directly optimize for positive interactions instead?



Criteria for Reinforcement Learning on Language Tasks

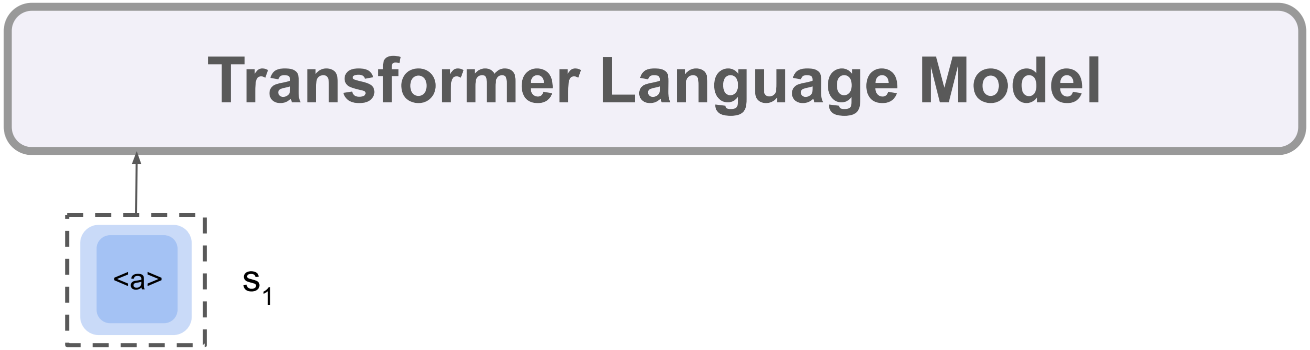
Method / Criteria	Easy to Use	Able to Optimize User Specified Rewards	Practical in Interactive Settings	Able to Leverage Existing Data	Temporally Compositional
Supervised Learning (BC)	✓	✗	✓	✓	✗
Filtered Fine Tuning (%BC)	✓	○	✓	✓	✗
Online RL	✗	✓	✗	✗	✓
ILQL (ours)	✓	✓	✓	✓	✓

Language Generation as a Token-Level POMDP

- The agent’s observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.

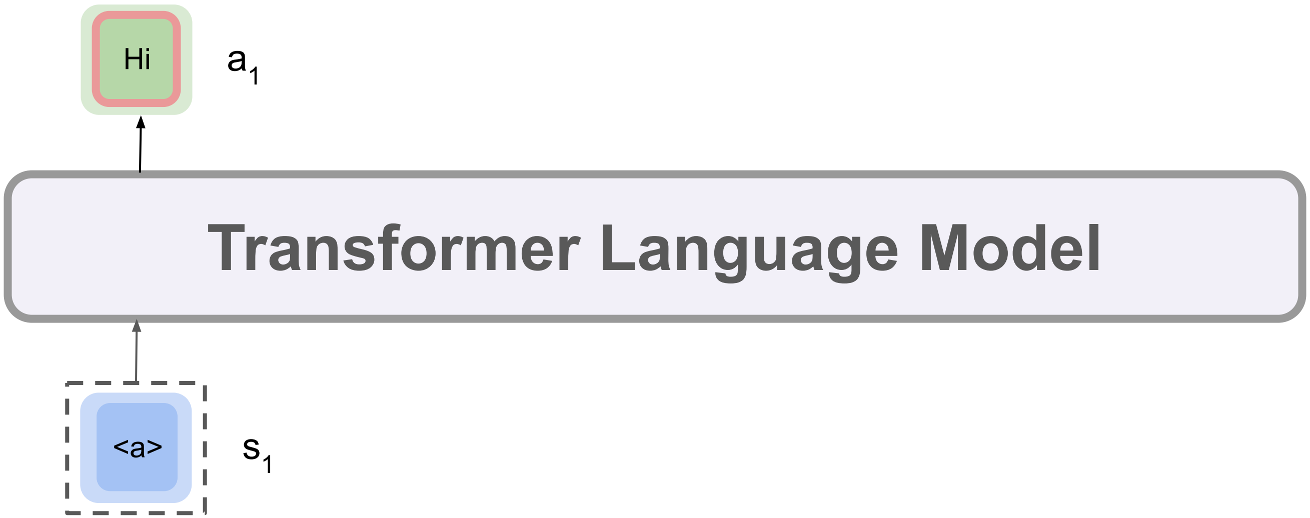
Language Generation as a Token-Level POMDP

- The agent’s observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



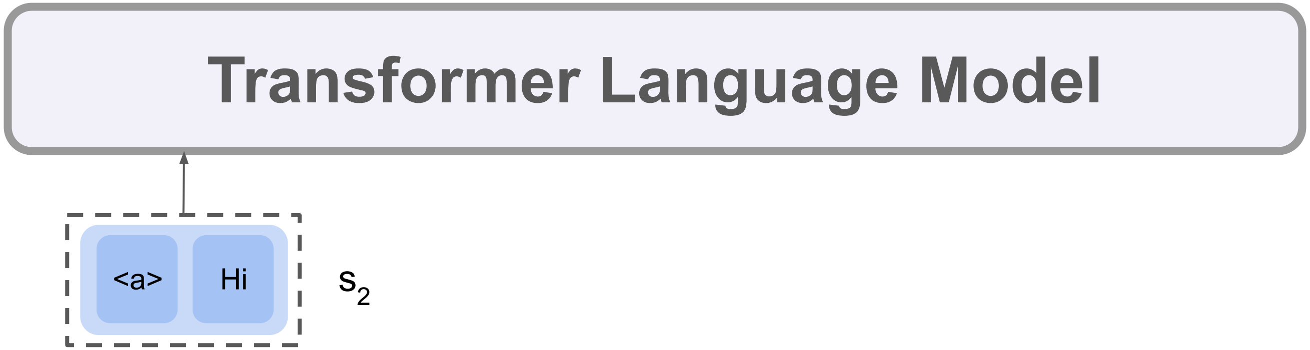
Language Generation as a Token-Level POMDP

- The agent’s observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



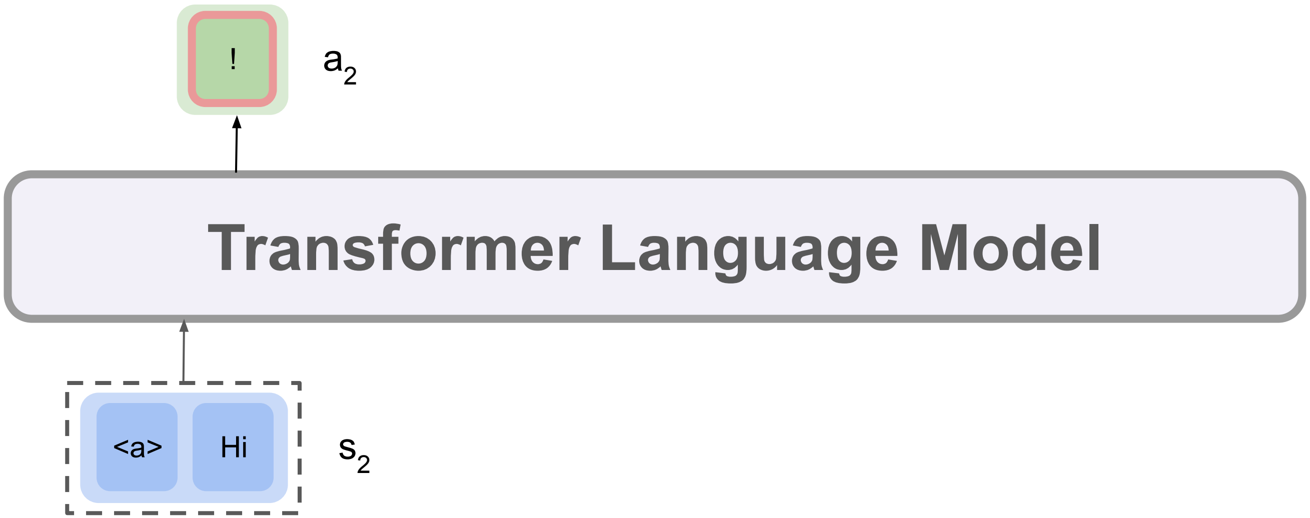
Language Generation as a Token-Level POMDP

- The agent’s observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



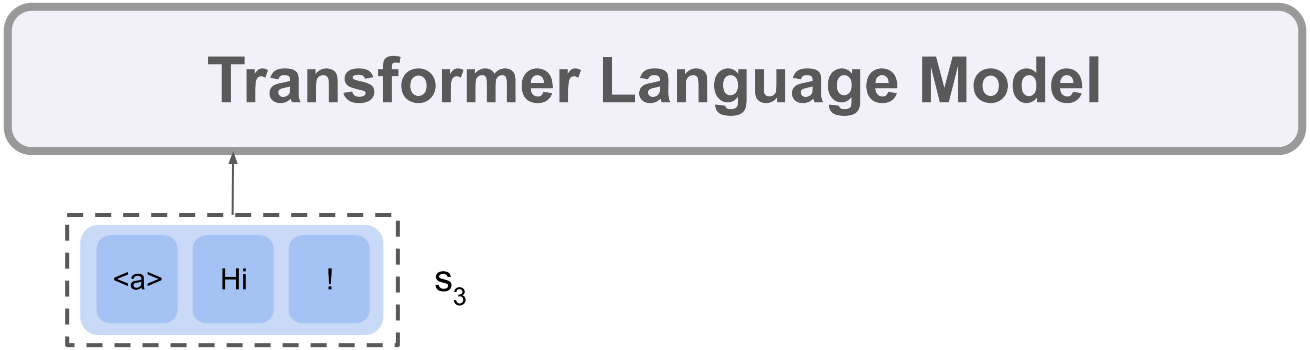
Language Generation as a Token-Level POMDP

- The agent's observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



Language Generation as a Token-Level POMDP

- The agent's observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.

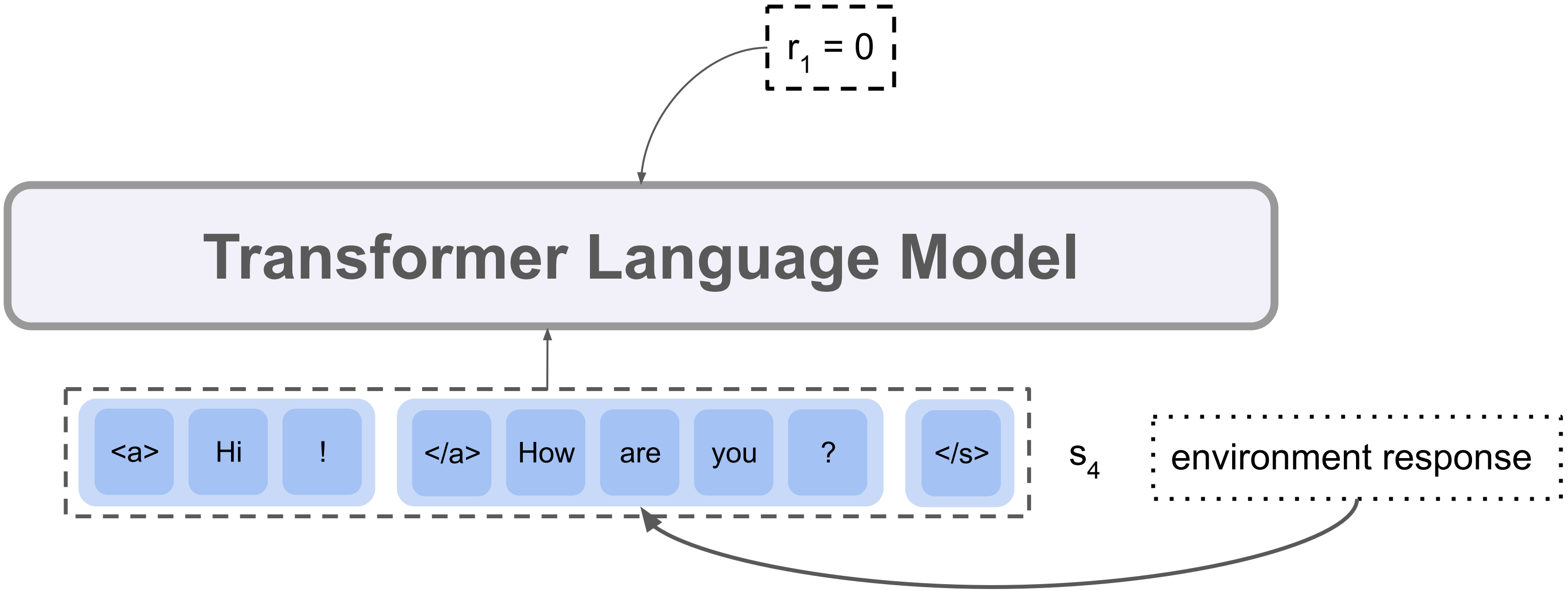


Language Generation as a Token-Level POMDP

- The agent's observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.

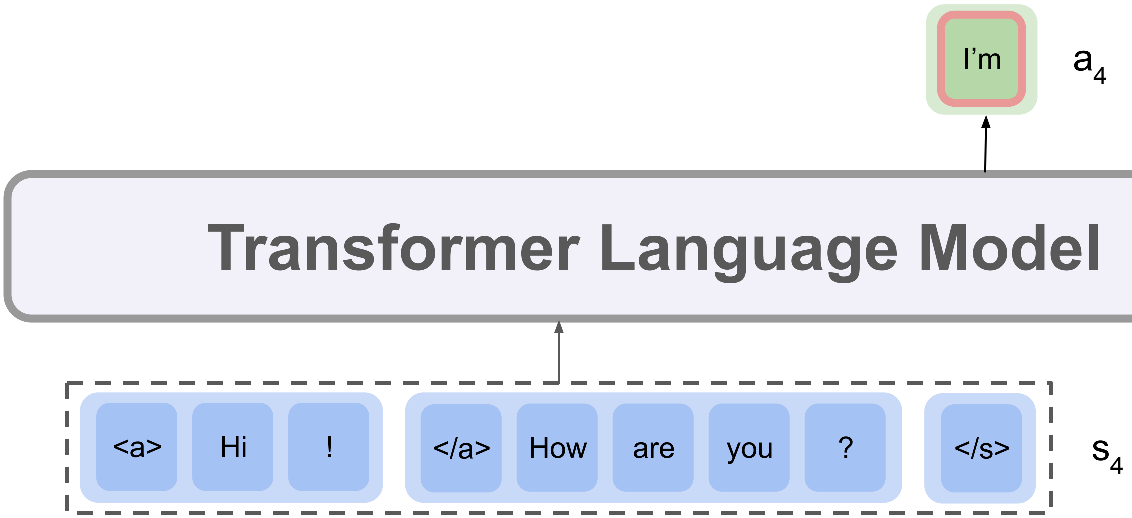
Language Generation as a Token-Level POMDP

- The agent’s observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



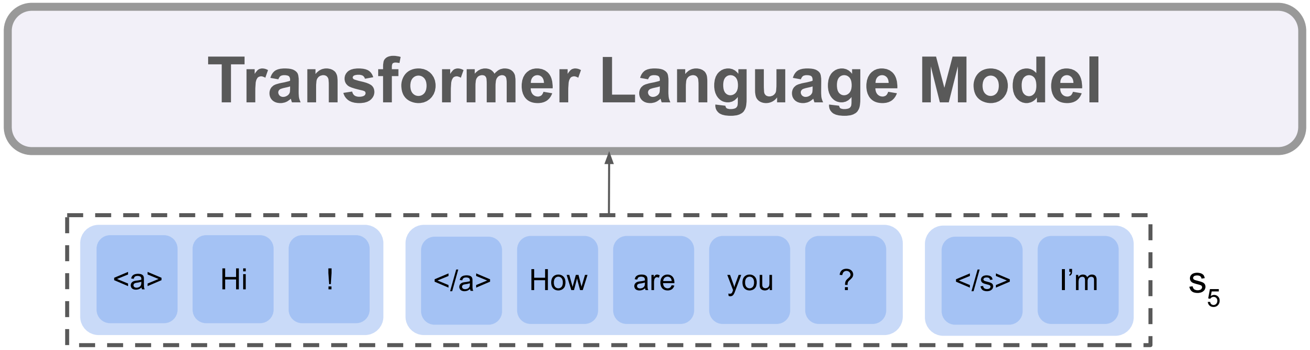
Language Generation as a Token-Level POMDP

- The agent’s observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



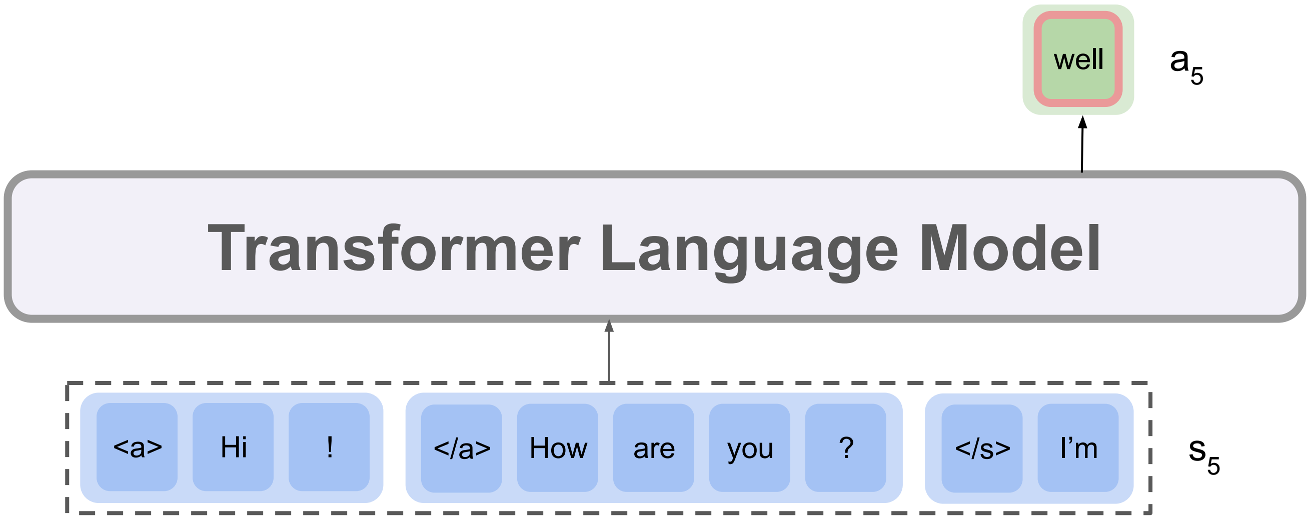
Language Generation as a Token-Level POMDP

- The agent's observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



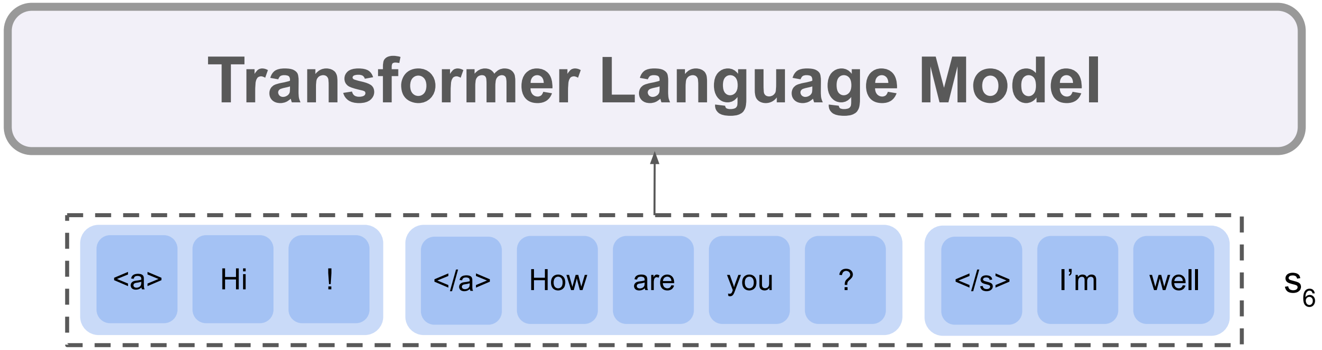
Language Generation as a Token-Level POMDP

- The agent's observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



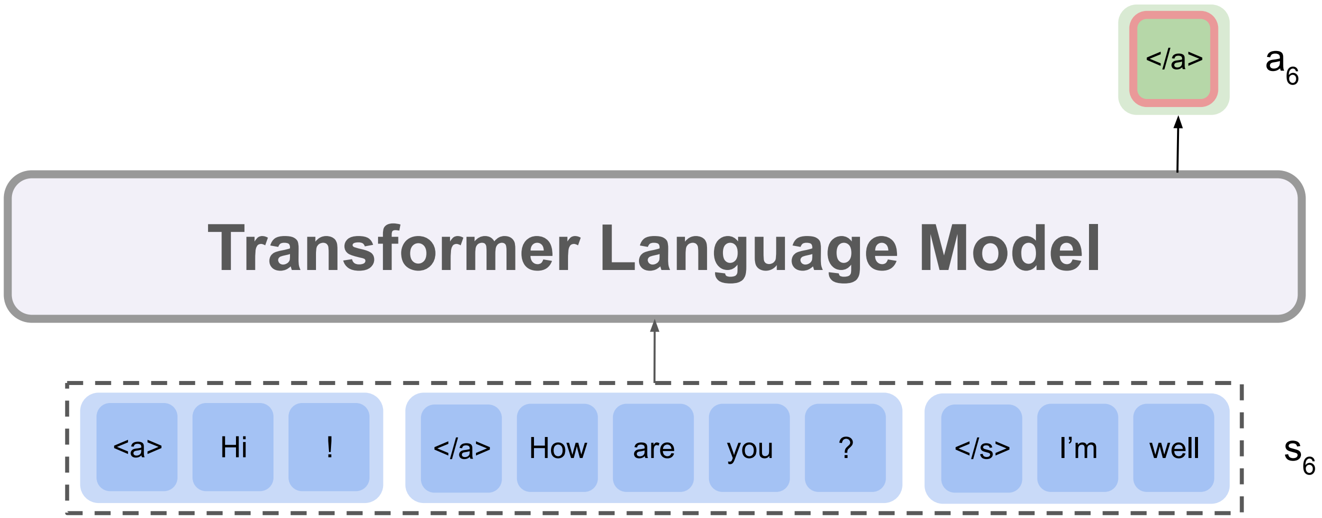
Language Generation as a Token-Level POMDP

- The agent’s observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



Language Generation as a Token-Level POMDP

- The agent’s observation is a history of tokens.
- The action space is the set of possible next tokens in the vocabulary.



POMDP Recap

The policy predicts the next token in an utterance given the history of past utterances.

$\pi($ Have a nice $) \longrightarrow$ day

POMDP Recap

The policy predicts the next token in an utterance given the history of past utterances.

π(Have a nice) → day

The environment responds to the agent.

T (Have a nice day ,) → You too !

POMDP Recap

The policy predicts the next token in an utterance given the history of past utterances.

π(Have a nice) ⟶ day

The environment responds to the agent.

T (Have a nice day ,) ⟶ You too !

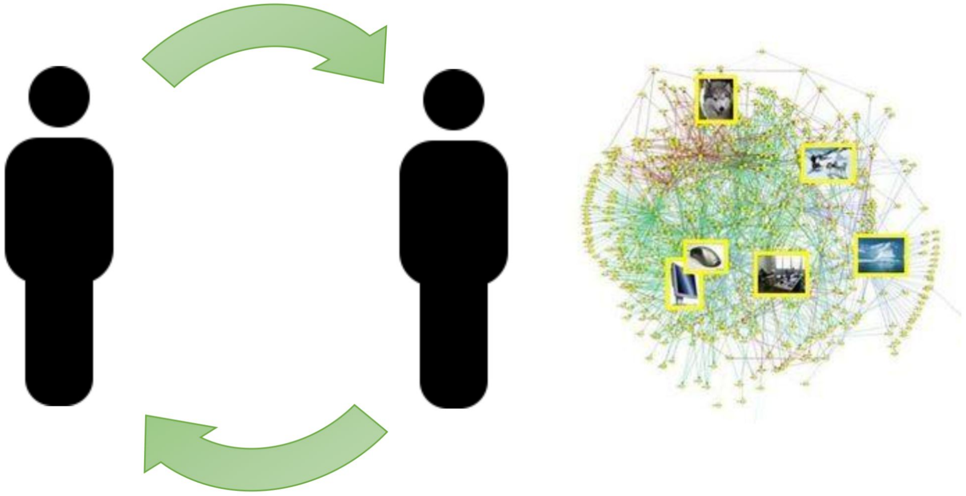
The agent gets rewarded at the end of each utterance.

r (Have a nice day ,) ⟶ **+1.0**

Value Function Learning

Lots of existing human-to-human dialogues on the internet

We can use offline-RL to extract optimal behaviors from this existing interactive data



Value Function Learning

Goal: learn a policy π which maximizes the task's expected cumulative reward: $\sum_{t=0}^{T} \mathbf{E}_{a_t,s_t}[r(s_t, a_t)]$.

Value Function Learning

Goal: learn a policy π which maximizes the task's expected cumulative reward: $\sum_{t=0}^{T} \mathbf{E}_{a_t, s_t}[r(s_t, a_t)]$.

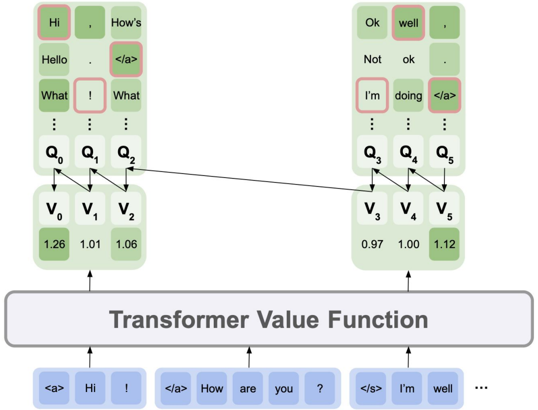
Constraint: we can only learn from a static dataset $\mathcal{D}$ of interactions and rewards collected by some potentially suboptimal "behavior policy", π_β.

Value Function Learning

Goal: learn a policy π which maximizes the task's expected cumulative reward: $\sum_{t=0}^{T} \mathbf{E}_{a_t,s_t}[r(s_t, a_t)]$.

Constraint: we can only learn from a static dataset $\mathcal{D}$ of interactions and rewards collected by some potentially suboptimal "behavior policy", π_β.

Method: learn value functions that represent the expected reward for the next token under the policy, and then choose the token that maximizes this value.



Implicit Q Learning

Implicit Q Learning: approximate the support constrained Bellman backup: $Q^*(s,a) = R(s,a) + \gamma \max_{a', \text{s.t. } \pi_\beta(a'|s')>0} Q^*(s',a')$

The in-support maximum is approximated by fitting a value function to an upper expectile of the Q function.

$$L_V(\psi) = \mathbf{E}_{(s,a)\ \sim D}[L_2^\tau(Q_{\hat{\theta}}(s,a) - V_\psi(s))]$$

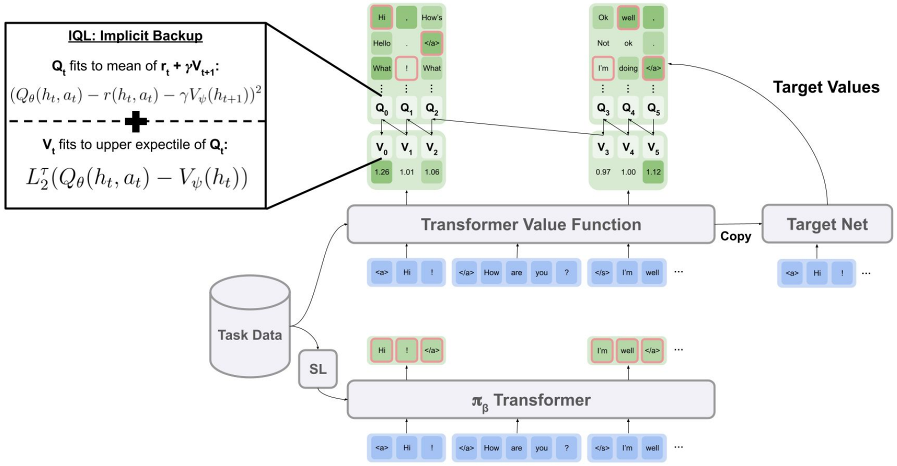
$$L_2^\tau(u) = |\tau - \mathbf{1}(u<0)|u^2$$

$$L_Q(\theta) = \mathbf{E}_{(s,a,s')\ \sim D}[(R(s,a) + \gamma V_\psi(s') - Q_\theta(s,a))^2]$$

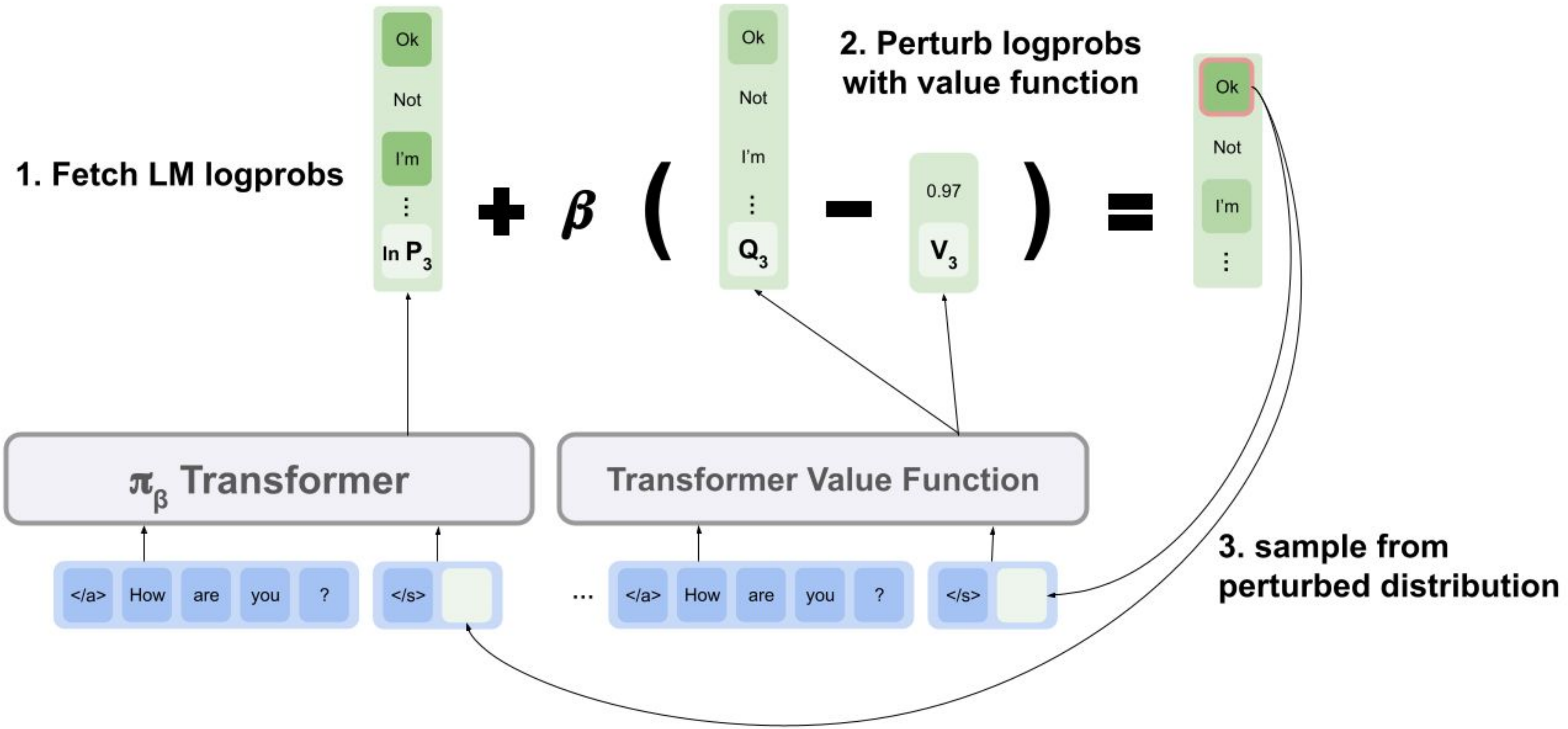
Implicit Language Q Learning – training

3 transformers:

1. Value function transformer (Q/V heads trained with IQL loss)
2. π_β transformer (standard supervised learning policy)
3. Target value function transformer (Polyak averaged copy of 1)

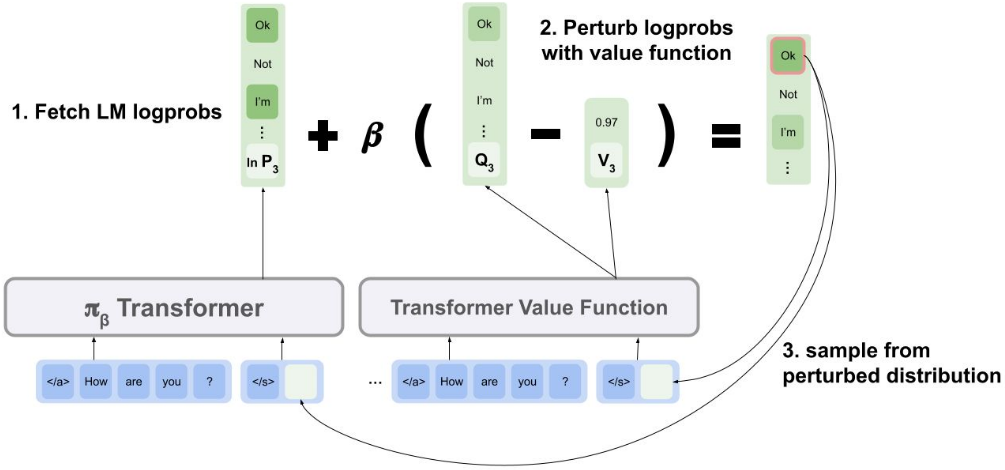


Implicit Language Q Learning – inference



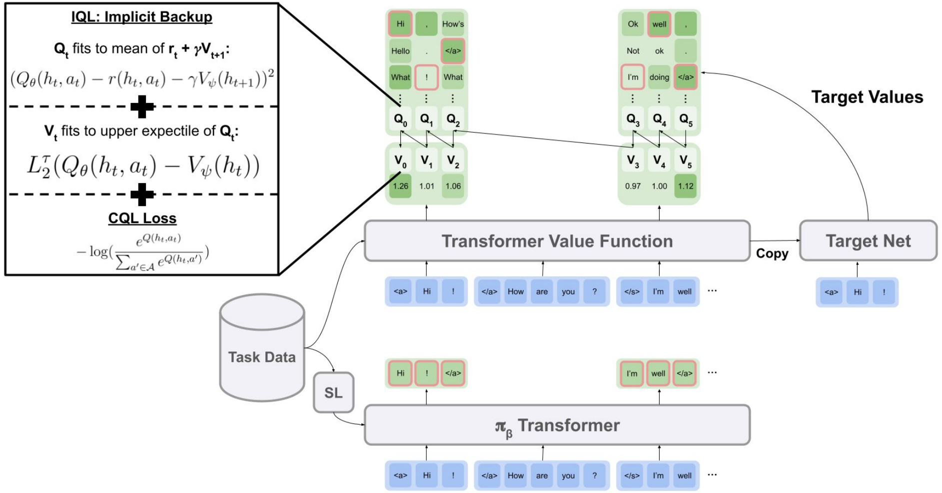
Implicit Language Q Learning – a problem with inference

- Q values for OOD actions can be arbitrarily large.
- LM doesn’t assign probability=0 to these OOD actions.
- The result is occasional OOD behavior.



Implicit Language Q Learning – two solutions

- We can fix this by either pushing down OOD probabilities or Q values.
  - Probabilities: add top-p filter or temperature to the logits
  - Q values: add NLL loss to the Q-values.
- Both work in practice. We find the latter typically requires the least amount of tuning.
- ILQL = IQL loss + CQL loss



Multi-Step Offline RL

- ILQL performs iterative policy improvement.

$$Q^*(s,a) = R(s,a) + \gamma \max_{a',\text{s.t. } \pi_\beta(a'|s')>0} Q^*(s',a')$$

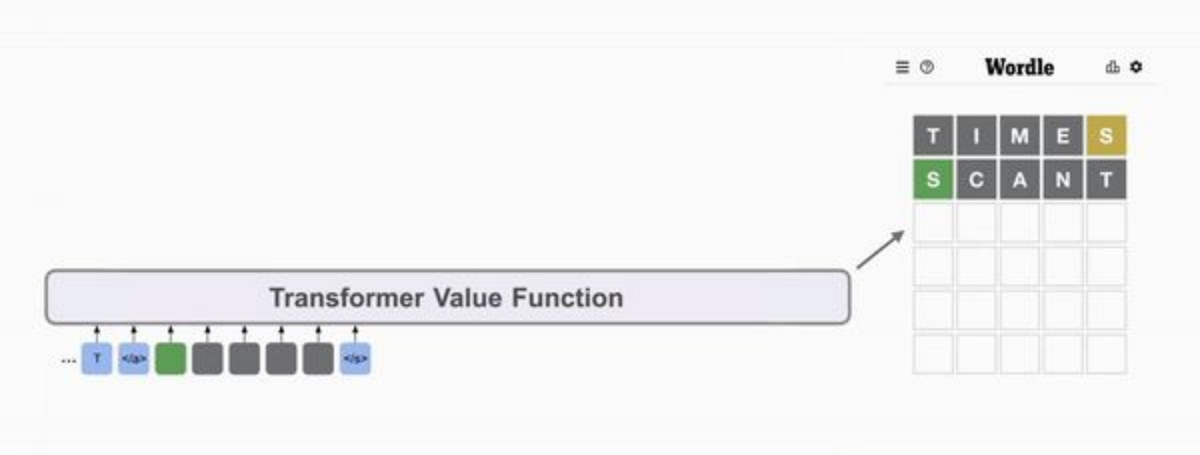
  - By fitting Q values to an approximate maximum over actions, we are recursively improving the policy.

- We expect ILQL to outperform methods which only perform a single step of improvement (SARSA).

1. Evaluate behavior policy: $Q_{\pi_\beta}(s,a) = R(s,a) + \gamma Q_{\pi_\beta}(s',a')$
2. Improve policy once: $\pi(s) = \max_a Q_{\pi_\beta}(s,a)$

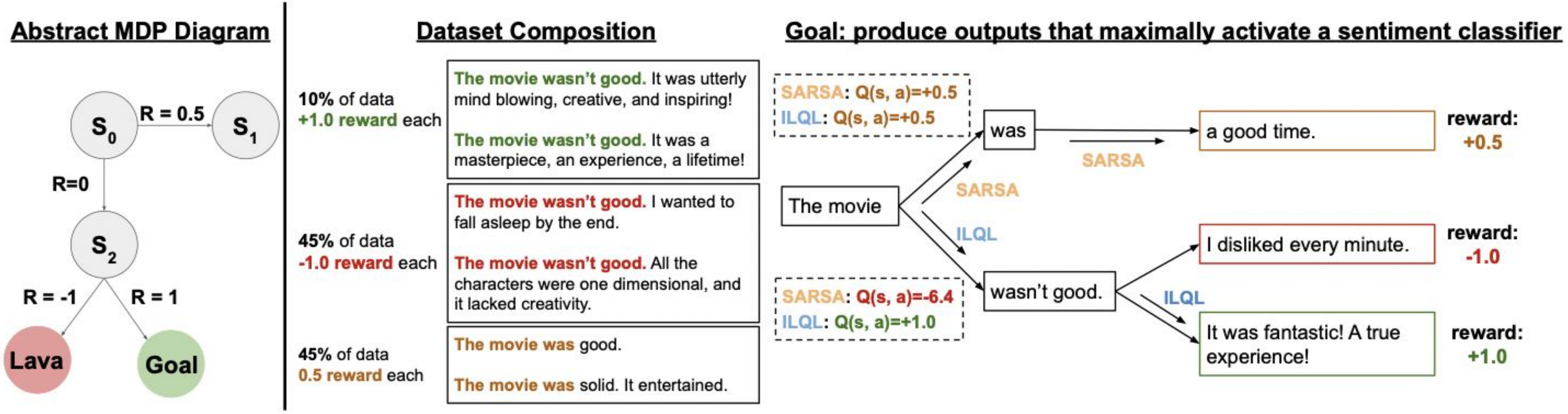
Proof of Concept: Multi-Step Offline RL on Wordle

- We present Wordle as an easy-to-use but challenging objective benchmark task to test offline RL algorithms.
- We use this task to test whether ILQL can perform multiple steps of policy improvement.



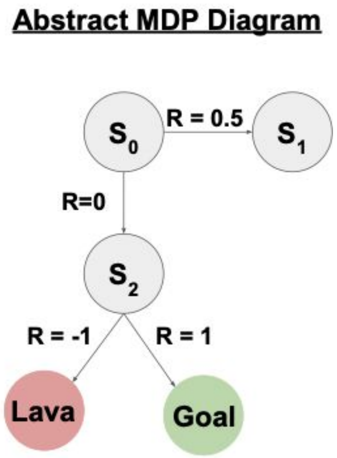
Proof of Concept: Multi-Step Offline RL on Wordle

- A notional example where we expect single step RL methods to catastrophically fail, and ILQL to succeed.
  - Good utterances tend to start with “The movie was...”
  - Bad utterances start with “The movie wasn’t...”
  - But the very best examples also start with “The movie wasn’t...”
- The data contains mostly suboptimal examples.
- Therefore, effective planning or multiple steps of policy improvement are needed to find the optimal policy.



Proof of Concept: Multi-Step Offline RL on Wordle

- To test ILQL's multiple steps of policy improvement, we instantiate this scenario in Wordle.
- We synthesize a dataset with trajectories from 3 different Wordle policies, each meant to represent one of the paths through the abstract MDP diagram.



π_{optimal}

represents S_0=>Goal
average reward: -2.647
~9% of data

```
arose      roast
halve      hoard
mange      board
You won!   You won!
reward: -2 reward: -2
```

$\pi_{\text{adversarial}}$

represents S_0=>Lava
average reward: -6.0
~45.5% of data
*repeats the first two words from optimal policy

```
shale      roast
short      drawn
shale      roast
shale      drawn
short      roast
short      drawn
You lost!  You lost!
reward: -6 reward: -6
```

$\pi_{\text{suboptimal}}$

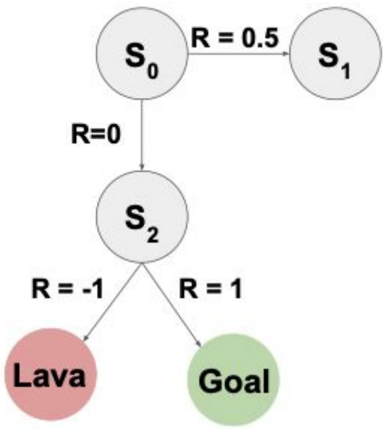
represents S_0=>S_1
average reward: -4.262
~45.5% of data

```
share      taker
rabbi      stony
hairy      fussy
aware      rider
paper      bunny
risen      You won!
You won!   reward: -4
reward: -5
```

Proof of Concept: Multi-Step Offline RL on Wordle

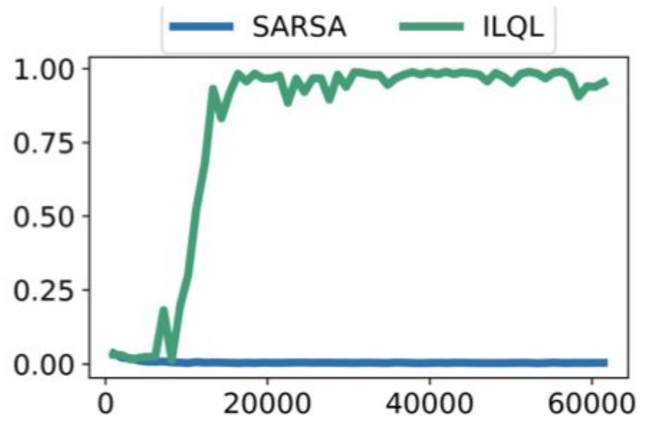
- ILQL assigns higher Q values to actions corresponding to paths to the “goal” state.
- SARSA assigns higher Q values to actions corresponding to paths towards the suboptimal $\mathbf{S_1}$ state.
- Confirming that ILQL can perform multiple steps of policy improvement.

Abstract MDP Diagram



We see a dramatic difference between ILQL and SARSA on this dataset.

Fraction of Q Values Where [S_0 => S_2] > [S_0 => S_1]



Wordle Tweets Data

- Does this finding about multiple steps of policy improvement transfer to more natural data distributions?
- We created a dataset of Wordle games scraped from Twitter.
- ILQL still outperforms single-step SARSA on this more realistic data distribution.

method	Wordle Score
ILQL	**-2.13** $\pm$ 0.03
SARSA	-2.23 $\pm$ 0.03
%BC	-2.38 $\pm$ 0.03
BC	-2.61 $\pm$ 0.03
π_{optimal}	-1.75 $\pm$ 0.02

Visual Dialogue Question Asking Task

- Can ILQL optimize different rewards in a complex dialogue setting?
- We use the Visual Dialogue dataset.
- Our agent asks questions about an image that it cannot see.
- The environment answers.
- 3 rewards: “standard”, “y/n”, “conservative y/n”.

Image Caption: *Tour buses are lined up on the street waiting for people.*
Questioner: how many buses?
Answerer: 2
Questioner: what color are buses?
Answerer: white and red
Questioner: how many people?
Answerer: 2
Questioner: what gender are people?
Answerer: 1 is male and 1 is female
Questioner: what are they wearing?
Answerer: 1 is wearing shorts and other is wearing shorts and shirt
Questioner: what color is their hair?
Answerer: dark brown

Visual Dialogue Quantitative Results

- ILQL outperforms baselines on most reward functions.
- ILQL finds good behavior even when the data is highly suboptimal.
- The optimal ILQL policy for one reward function is not optimal for others.

method	standard	y/n	conservative y/n
ILQL	-5.22 ± 0.13	**-5.69**±0.13	**-6.57** ± 0.18
SARSA	-5.14 ± 0.13	-6.19±0.15	-7.77 ± 0.20
%BC	-5.07 ± 0.13	-7.48 ± 0.21	-9.13 ± 0.22
BC	-5.25 ± 0.13	-10.85 ± 0.27	-15.16 ± 0.35

train/eval	standard	y/n	conservative y/n
standard	**-5.22** ± 0.13	-11.12 ± 0.30	-14.97 ± 0.36
y/n	-5.41 ± 0.12	**-5.69** ± 0.13	-8.24 ± 0.22
conservative y/n	-5.29 ± 0.13	**-5.42** ± 0.13	**-6.57** ± 0.18

Visual Dialogue Qualitative Results

Qualitatively, models trained on rewards which penalize yes/no questions ask fewer such questions.

Standard Reward Function:

Image Caption: *A man, with other people in the background, is sitting at a picnic table with a cell phone in front of him while he uses a fork and knife to cut a dessert on a plate.*
Questioner: is man young?
Answerer: yes
Questioner: is he wearing glasses?
Answerer: no
Questioner: is there food on table?
Answerer: yes
Questioner: is there food on table?
Answerer: yes

Yes/No Penalty:

Image Caption: *A person flying a kite over a sand dune.*
Questioner: what color is kite?
Answerer: it's black and white
Questioner: is person male or female?
Answerer: can't tell
Questioner: what is person wearing?
Answerer: i can't tell
Questioner: what color is sand?
Answerer: dark brown
Questioner: is person flying kite flying in air?
Answerer: yes
Questioner: what color is kite?
Answerer: black

Conservative Yes/No Penalty:

Image Caption: *Tour buses are lined up on the street waiting for people.*
Questioner: how many buses?
Answerer: 2
Questioner: what color are buses?
Answerer: white and red
Questioner: how many people?
Answerer: 2
Questioner: what gender are people?
Answerer: 1 is male and 1 is female
Questioner: what are they wearing?
Answerer: 1 is wearing shorts and other is wearing shorts and shirt
Questioner: what color is their hair?
Answerer: dark brown

Reddit Comments Task

- Offline RL can optimize in the face of environment noise.
- Can ILQL optimize its generations of maximally diverse open-domain text when subjected to highly stochastic reward functions based on subjective human judgement?
- We test this using a large dataset of 4 million Reddit comments.
- Optimize agents for two different subjective reward functions:
    1. Generating non-toxic comments (“toxicity”).
    2. Generating positive upvote comments (“upvotes real” and “upvotes model”)

Reddit Comments Results

method	toxicity	upvotes real	upvotes model
ILQL	**0.0**±0.0	**9.83**±0.04	**10.0**±0.0
SARSA	**0.0**±0.0	6.23±0.15	**10.0**±0.0
%BC	-0.74±0.07	7.06±0.14	7.86±0.13
BC	-3.51±0.13	4.87±0.16	4.87±0.16

- ILQL obtains the maximum reward on two of the three rewards.
- Fineuning on only non-toxic or positive upvote comments sometimes generates undesirable outputs.
- ILQL is able to more robustly optimize these more subjective, higher-variance reward functions.

Reddit Comments Results

ILQL per-token advantages for toxic comments generated by filtered finetuning model

advantage:	0.1	-0.1	-0.9	-0.5	0.1	0.2	0.0	-0.9	-0.3	-1.1
token:	And	they	censor	your	comments	on	this	horrible	site	.

The learned value function assigns a lower advantage to negative words.

Abalations

method	max score	σ w.r.t hparams
ILQL	**-5.69**$\pm$0.13	0.42
CQL	-7.32$\pm$0.17	1.98
ψ	-10.05$\pm$0.18	0.60
SARSA	-6.19 $\pm$ 0.15	**0.27**
DT	-6.70 $\pm$ 0.17	1.15
ILQL (AWR)	-5.96$\pm$0.13	2.82
%BC	-7.48 $\pm$ 0.21	0.72
BC	-10.85 $\pm$ 0.27	-

- We abalate the choice of Offline-RL algorithm on the Visual Dialogue “y/n” reward.
- ILQL outperforms prior offline RL methods applied to language models.

Conclusion

- ILQL can be used to …
  - optimize language models over multi-turn, interactive dialogue tasks.
  - Learn from diverse open-domain text
- We look forward to future work on advancing RL algorithms for interactive language tasks.