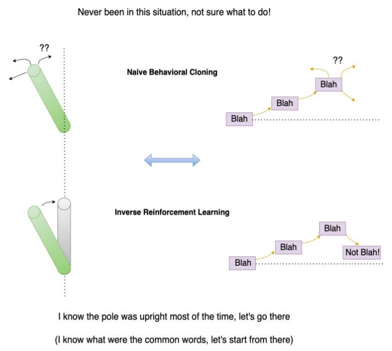
Natural Language Generation by Adversarial-Free Imitation Learning

Anuj Nagpal

ICME, Stanford University

Problem

- Despite their impressive performance, current language generation models are prone to **autoregressive bias** issue wherein once they start generating text that is slightly different from the training data, they continue generating even weirder data with lots of repetitions, leading to a feedback loop.
- A parallel can be drawn here with **naive behavioral cloning** approach in reinforcement learning where once an agent reaches a state expert hasn't seen, the agent can keep drifting away from demonstrated states due to error accumulation.


Never been in this situation, not sure what to do!
??
Naive Behavioral Cloning
??
Blah
Blah
Blah
Inverse Reinforcement Learning
Blah
Blah
Not Blah!
Blah
I know the pole was upright most of the time, let's go there
(I know what were the common words, let's start from there)

Proposed Solution

- Frame the task as an **Inverse RL** problem to learn a reward function under which the expert's trajectory is optimal.
- Agent learns the high density states so that it can recover on encountering **out of distribution events**.
- Major issue with traditional Inverse RL approaches is that they use rewards in some form via **adversarial training which can be unstable**.
- Solution - Learn a Q-function directly from expert demonstrations to obtain an imitation policy. **No rewards or adversarial training required!**

Technical Approach

- **NLP to RL Formulation**: We use the **BERT [CLS] encoding** of a text sequence to represent state and **vocabulary size as the action space**. This is a deterministic discrete control problem where the agent will be learning in an online fashion.
- Our model learns a Q-function from expert trajectories by **Inverse-Q Learning** via a Double Deep Q-network and recovers agent's policy.

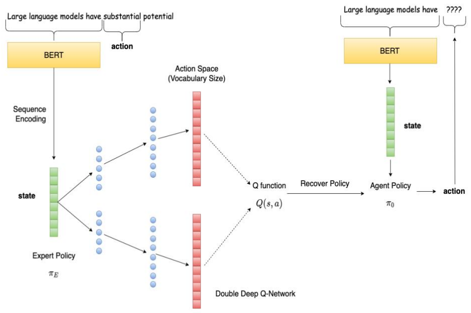


Figure 1. Inverse Q-Learning for Language Generation

- Expert trajectories are created by breaking down sentences in wikitext training split into (state, action, next state) or (s, a, s') tuples. A **replay buffer** is maintained for both expert and agent to sample these experiences for learning.
- To ensure diversity across the high dimensional action space, we use a **softmax temperature** that can be learned to match a target entropy.

Optimization Objective

$$\max_Q \mathcal{J}(Q) = \mathbb{E}_{(s,a,s')\sim\text{expert}}[Q(s,a) - \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)} V^\pi(s')]$$

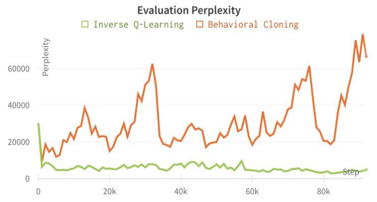
$$-\mathbb{E}_{(s,a,s')\sim\text{(agent, expert)}}[V^\pi(s) - \gamma V^\pi(s')]$$

$$-\frac{1}{4\alpha}\mathbb{E}_{(s,a,s')\sim\text{(agent, expert)}}[(Q(s,a) - \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)} V^\pi(s'))^2]$$

Experimental Results

We trained our agent with Behavioral Cloning (BC) as well as Inverse Q-Learning (IQL) for roughly 2000 episodes of 50 token length sequences and compared the **model's perplexity on wikitext test split tokens** at regular intervals as plotted below:

Wikitext Evaluation Perplexity - BC vs IQL



Example Generated Sequence

it met with positive sales in japan, and was praised by both japanese and western critics. after release, it received downloadable content, along with an expanded edition in november of that year. it was also adapted into manga and an original video animation series.

We observe **lower evaluation perplexity for IQ-Learning** as compared to BC across multiple temperature and replay memory size configurations:

Temperature	Memory Size	Perplexity (BC)	Perplexity (IQL)
1	500	34.78e2	**16.67e2**
1	5000	182.98e2	**63.74e2**
1	50000	788.59e2	**42.96e2**
2	50000	118.55e2	**44.11e2**
3	50000	70.88e2	**39.05e2**

Conclusion

- We were able to show that Inverse Q-Learning is able to outperform Naive Behavioral Cloning with a simple model, which acts as a good proof-of-concept to **plug the loss objective from our method into a big language model's (like GPT) training procedure** for better performance.

- Special thanks to Chris Cundy and Div Garg