Using RL for Non-Greedy Dependency Parsing

Sidhart Krishnan, Arvind Saligrama
Department of Computer Science, Stanford University

Problem

- **Motivating Problem:** Use reinforcement learning to perform non-greedy decoding for transition-based parsers
- Dependency relationships can improve performance on a variety of NLP tasks and so improving dependency parsing is important
- Supervised methods perform **greedy decoding**
  - RL could be useful because it considers future reward and thus their policies are non-greedy.

Background

Problem Setup: Create an RL Agent that predicts the next move in a transition-based dependency parser and aims to maximize the unlabeled attachment score (UAS)

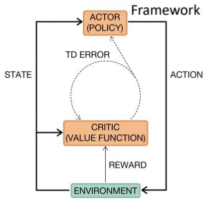
A Fast and Accurate Dependency Parser Using Neural Networks (Chen and Manning [1], 2014): The authors use a neural network to determine the next transition

Dependency Parsing with Deep Reinforcement Learning (Shen et al. [2], 2016): The authors aim to build a reinforcement-based dependency parser to perform non-greedy decoding.

Methods

- Transition-based dependency parsing aims to create a dependency structure for a sentence. We specifically used the arc standard transition system.
- We create a parser environment for our RL setup
  - Is possible because we can frame the shift-reduce parser as a Markov Decision Process (MDP)
  - We must frame the environment as a "game" with a reward function that the agent aims to maximize
- Tested two actor-critic RL algorithms on the parser environment: **A2C** and **PPO**
  - Actor-critic methods have a policy network which decides actions and a value network to determine the expected future reward
  - A2C and PPO differ in how the loss is calculated

Actor Critic Framework

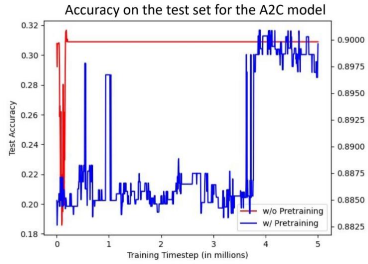


Experimental Setup

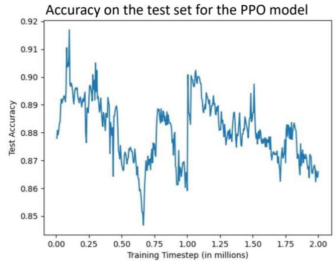
- **Task:** Used the English Penn Treebank (PTB) dataset to load in examples to the environment
  - Agent returns an action based on current parse of sentence
- **Metric:** Unlabeled Attachment Score (UAS) – the percentage of tokens that have the correct head
- The policy network for the A2C and PPO parsers were both initialized to match the parameters of neural dependency parser from Chen and Manning
  - Also trained an A2C model w/o supervised pretraining to determine if better than random policy (UAS = 12.90)

Results

Accuracy on the test set for the A2C model



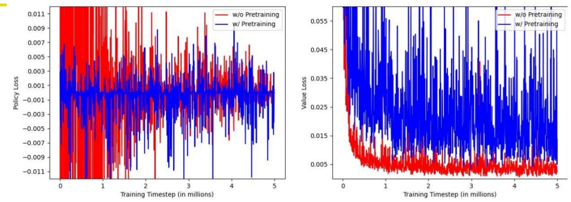
Accuracy on the test set for the PPO model



Model	Avg UAS
Baseline	88.88
A2C	31.20
A2C*	**89.19**
PPO*	89.12

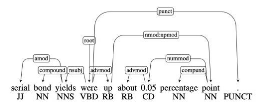
* Indicates that model was pretrained with supervised weights

Loss for Policy and Value Networks during A2C Training

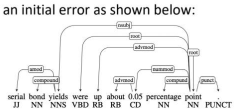


Analysis

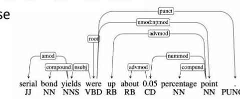
- Policy loss for pretrained A2C much smaller than policy loss of non-pretrained
- Value loss for pretrained A2C initially much higher as critic network must catch up to pretrained policy network
- The RL models often performs better after an initial error as shown below:



Gold Parse



Supervised Parse (UAS = 50)



A2C/PPO Parse (UAS = 80)

Conclusions

- The A2C and PPO models w/ pretraining performed slightly better than the supervised model on the test data
- Initializing the parameters with a pretrained supervised model was critical for the RL model to properly explore the space and learn

Acknowledgements

We would like to thank Allan Zhou for his guidance and patience throughout this project.

References

[1] Chen, Danqi and Manning, Christopher. A Fast and Accurate Dependency Parser using Neural Networks. 2014 Conference on Empirical Methods in Natural Language Processing, 29 Oct. 2014.
[2] Shen, Ying, et al. Dependency Parsing With Deep Reinforcement Learning. 29th Conference on Neural Information Processing Systems, 2016.