Evaluating RL-based Approaches for Knowledge Graph Reasoning

Parthasarathy Suryanarayanan, Hengkai Qiu, David Uvalle
Department of Computer Science, Stanford University

Link to Video: https://www.youtube.com/watch?v=yDI5xSQi1wA

Objective

The objective of this project is to evaluate the reinforcement learning methods for knowledge graph [KG] reasoning. As part of this work, we intend to

- **To develop an efficient implementation of** the method presented in the paper "DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning" [1]
- **Further investigate the model hypotheses**, in order to improve the performance, especially the reward functions, policy network architectures and other potential techniques

Background & Social Impact

- **KGs**, as a type of knowledge representation, have gained much attention in natural language processing in recent years. KGs can effectively organize and represent knowledge so that it can be efficiently utilized in advanced applications such as question answering and recommender systems.
- **Reasoning over KGs** is an active research topic, since it can obtain new knowledge and conclusions from existing data. The typical methods are either rule-based or embedding-based. The paper approaches the reasoning problem from a reinforcement learning perspective which is both novel and intriguing.

Dataset

- **The dataset consists of two files of triples** that are subsets of a larger dataset. One is the FB15K-237[2] as part of FB15K and a secondary is the NELL-995 that was created as part of the paper by filtering the original NELL[3] dataset.
- **In order to create the NELL-995**, the authors selected only triples with top-200 relations; also added inverse triples, so the agent was able to step backward in the KG.
- **some examples** for the triple (h, r, t) are:
  (Donald Graham, works_for, Company TNT), (Band of Brothers, written_by, Graham Yost), (Tom Hanks, person_languages, English).

Dataset	# Entities	# Relations	# Triples
FB15K-237	14,505	237	310,116
NELL-995	75,492	200	154,213

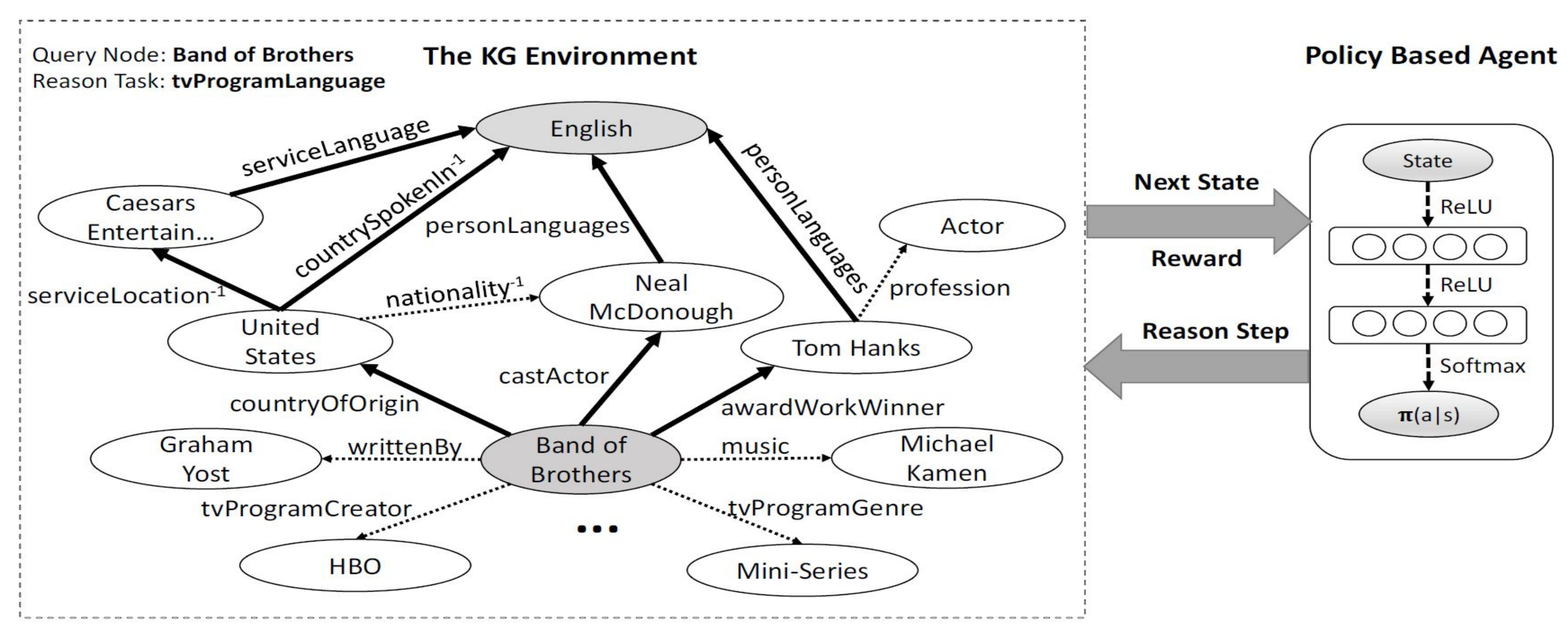
Reinforcement Learning Framework

Reinforcement learning for path learning between entities. The KG is modeled as a partially observed MDP, which is a 4-tuple **<S,A,P,R>**.

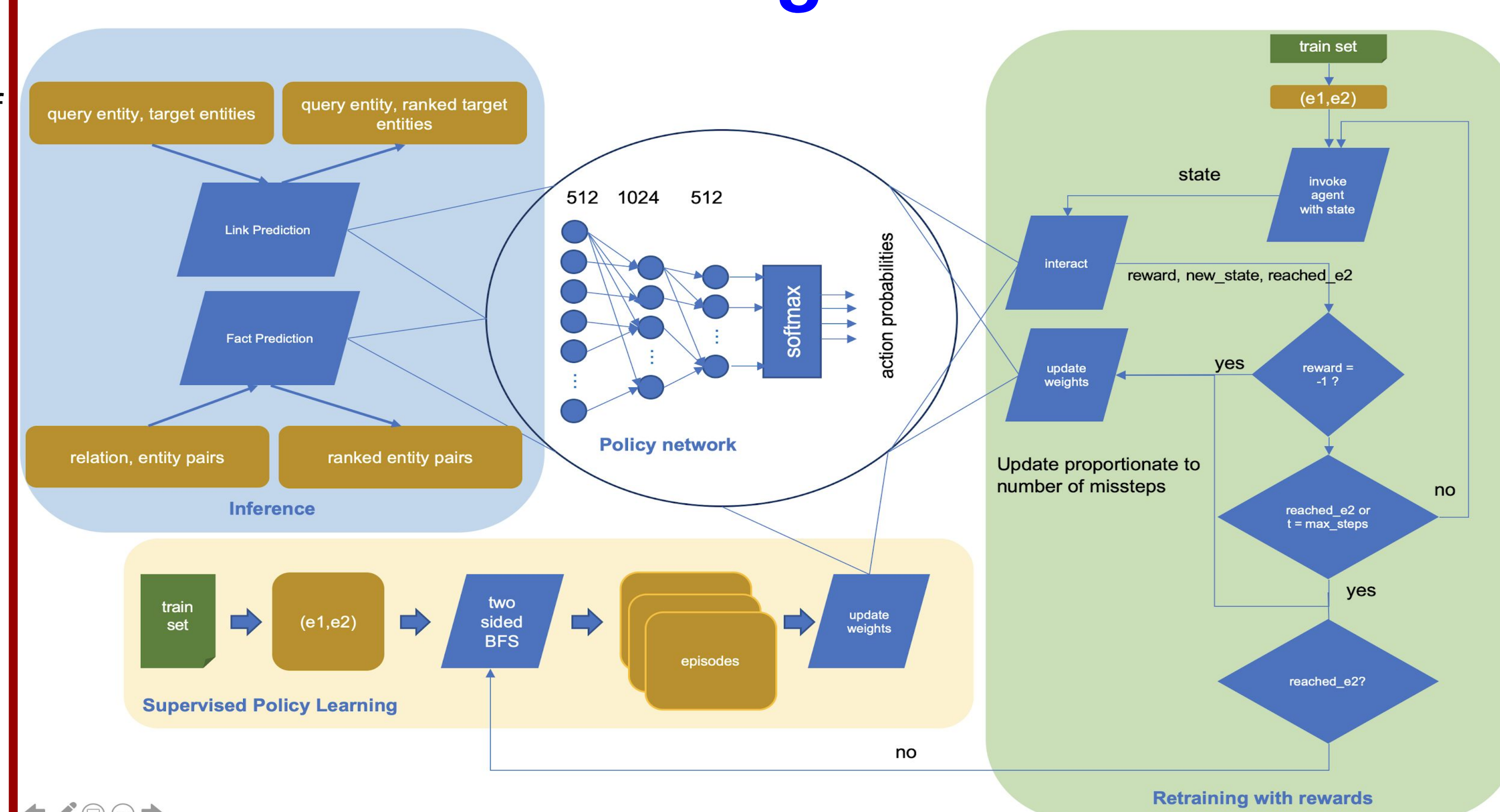
The RL agent is represented as a **policy network** $\pi(s;\theta) = p(a|s;\theta)$ which maps the state vector to a stochastic policy.

- ***States***: $s_t = (e_t, e_{target} - e_t)$ where e_t denotes the embeddings of the current entity and e_{target} s the embedding of the target entity. The paper uses translation-based embeddings to represent entities and relations.
- ***Actions***: the relation picked to extend its path at each step until it reaches the target entity, the action space is defined as all the relations in the KG.
- ***Reward Function***: the reward function is a composition of factors designed to encourage the agents to find the predictive paths, including Global Accuracy, Path Efficiency and Path Diversity.

$$r_{\text{GLOBAL}} = \begin{cases} +1, & \text{if the path reaches } e_{target} \\ -1, & \text{otherwise} \end{cases} \qquad r_{efficiency} = \frac{1}{length(p)} \qquad r_{diversity} = -\frac{1}{|F|}\sum_{i=1}^{|F|} cos(p, p_i)$$



Model Training & Inference



Experiment Results, Discussion

- The original implementation has several issues - severe functional bugs and extremely poor training speeds due to inefficient implementation which makes us wonder if the final implementation used by the paper was released at all.
- We spent several days in fixing these bugs and implemented numerous speed improvements that has resulted in 15x speed increase in the training pipeline so far. These include identification and mitigation of severe I/O contention issues and addition of catching results of the BFS etc.
- However, this is still not sufficient for quick experimentation as it currently takes 55 hours for a complete training run.

Future Work and Citations

As the next steps, we further plan to optimize the training pipeline If we are successful in optimizing the speed, we plan to:

- Experiment with replacing reward factor that penalizes very hard large paths with a linear function.
- Experiment with different hyperparameter settings such as the weights of the different reward functions.
- Experiment with expanding policy network architecture

[1] Xiong et al., Deep path: A reinforcement learning method for knowledge graph reasoning, EMNLP 2017. arXiv:1707.06690
[2] Representing Text for Joint Embedding of Text and Knowledge Bases, Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, Michael Gamon, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing
[3] Toward an architecture for never-ending language learning, Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, Jr., Tom M. Mitchell, AAAI'10 Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence