Neuro-Symbolic Techniques for Description Logic Reasoning (Student Abstract)

Gunjan Singh,[1] Sutapa Mondal,[1] Sumit Bhatia,[2] Raghava Mutharaju[1]
[1]Knowledgeable Computing and Reasoning (KRaCR) Lab, IIIT-Delhi, India,
[2]IBM Research AI, New Delhi, India
{gunjans, sutapa18056, raghava.mutharaju}@iiitd.ac.in, sumitbhatia@in.ibm.com

Abstract

With the goal to find scalable reasoning approaches, neuro-symbolic techniques have gained significant attention. However, the existing approaches do not take into account the inference capabilities of ontology languages that are based on expressive description logic (such as OWL 2). To fill this gap, we propose two approaches: an ontology-based embedding model for theories in $\mathcal{EL}^{++}$ description logic and a reinforcement learning-based solution for efficient tableau-based reasoning on description logic. We describe promising initial results of our efforts towards these directions and lay down the direction for future work.

Introduction

Motivated by the need for reasoning approaches that can scale well even on the most expressive, and large ontologies, neuro-symbolic approaches have received major attention in recent times. The idea is to combine the robustness and learning capabilities of the artificial neural networks along with the precise reasoning abilities of logic-based approaches. Recent papers by (Hitzler et al. 2019) and (d' Amato 2020) briefly identify some of the key challenges and briefly discuss the potential of neuro-symbolic reasoning approaches when compared with traditional reasoning approaches.

In relation to this theme, we discuss two approaches that we are currently working on: an ontology-based embedding model for theories in $\mathcal{EL}^{++}$ description logic and a reinforcement learning-based solution for efficient tableau-based reasoning for description logic. To the best of our knowledge, most of the existing work in these directions, mainly cover tasks in the context of knowledge graph (KG) completion and relatively little attention has been paid to utilizing the inference capabilities of schema defined in ontologies. Further, many of the existing approaches do not take into account the constraints and characteristics of underlying ontology. We describe promising initial results of our efforts towards these directions and lay down the direction for future work.

Embedding Model for $\mathcal{EL}^{++}$ Description Logic

We propose an ontology embedding model (EmEL^{++}) as a non-traditional way of performing reasoning tasks. In our first step towards ontology embedding we looked at OWL 2 EL profiles which have lot of applications in biomedical domain. Our proposed approach is similar to embedding model (ElEm) by Kulmanov et al. (2019). Although, Kulmanov et al. (2019) focused on a link prediction task for protein interaction network, our work is more challenging as it addresses the ontology which deals with more complex relations compared to KGs. In $\mathcal{EL}^{++}$, every reasoning task can be reduced to subsumption task (Baader, Brandt, and Lutz 2005). Thus, in this work, we evaluate the performance of embeddings on a subsumption reasoning task. Further, an $\mathcal{EL}^{++}$ ontology can be reduced to normal forms to represent the complex associations in a simpler way. The model captures geometric interpretations of classes and relation in vector space wherein, it visualizes the classes and relations as n-balls and translational vectors respectively. The approach is represented as an optimization problem where, for each of the associated constructs in $\mathcal{EL}^{++}$ a loss function is defined which is not the case with (Kulmanov et al. 2019), the objective of the problem is to minimize the losses associated to all the constructs so as to preserve the relationships between the classes in vector space. The training phase involves a neural network based linear model wherein each input node is associated to the obtained normal forms. The model uses two parallel embedding layers, one for the classes and another for the relations to map them to n-dimensional vector space while minimizing the loss. We define accuracy metric to evaluate embeddings which represents the correct predictions for subsumption reasoning task such that the n-ball of subclass is subsumed in n-ball of superclass. We carry out experiments on multiple datasets with varying characteristics. The initial results showed that our model was able to capture the relationship of classes and relations well in vector space. Table 1 draws a comparison with the recent work on $\mathcal{EL}^{++}$ wherein, it shows accuracy over training, testing and inferences data. The inferences data is curated using a standard reasoner ELK on training data. The quality of embeddings is then evaluated over these data. Overall, the EmEL^{++} embeddings resulted in a decent performance to recognize the inferences. Thus, it looks promising for further research in this direction.

	Training		*Testing*		*Inferences*	
	ElEm	EmEL^{++}	ElEm	EmEL^{++}	ElEm	EmEL^{++}
GALEN	0.27	**0.64**	0.20	**0.53**	0.27	**0.64**
GO	0.45	**0.59**	0.35	**0.44**	0.48	**0.62**
ANATOMY	0.09	**0.48**	0.07	**0.22**	0.10	**0.49**
SNOMED	0.24	**0.55**	0.18	**0.34**	0.22	**0.48**

Table 1: Accuracies achieved by the ElEm and EmEL++ embeddings in terms of geomteric interpretation of the classes in different ontologies.

Reinforcement Learning Based Solution for Efficient Tableau Reasoning

Let C be a concept and I be an interpretation. The aim of the tableau algorithm (Möller and Haarslev 2009) is to generate a finite interpretation I such that $C^I \neq \emptyset$. Without loss of generality, we assume all concept descriptions are in negation normal form (NNF). Now, we start with a knowledge base, A:= C(x), and apply tableau expansion rules sequentially. This leads to a form of a search tree (Figure 1), with the root node C(x). Here, the edges represent the applied tableau rules. This tree will go as deep as to a point where no further rules are applicable or an obvious contradiction is detected. We call our knowledge base A as satisfiable if and only if one of the complete ABoxes is open, i.e., does not contain an obvious contradiction of the form $\{A(x), \neg A(x)\}$. These rules may be deterministic (such as conjunction) as well as non-deterministic (such as disjunction). The second type of rules might lead to the wrong track and would require to backtrack. Thus, with the increase in either the size or expressivity of the knowledge base, the tree becomes broader and deeper and hence the reasoning complexity increases exponentially. To deal with this kind of non-determinism, our endeavor is to propose a reinforcement learning solution that can cash in on the exploration-exploitation trade-off of the reinforcement learner and reduce its time complexity.

We define our Reinforcement Learning framework as a discounted Q-Learning model. Q-Learning is a model-free algorithm that learns an action policy over time by maximizing the expected cumulative reward function. So, we have a finite set of states $\mathbf{S}$ (represented by the nodes in our tree), a finite action space $\mathbf{A}$ (represented by the edges in our search tree) and a numerical score called as a reward $\mathbf{R}$ (received by the agent for a state-action pair). The weight for a step from a state, Δt steps into the future is calculated as $\gamma^{\Delta t}$ where γ is the discount factor ($0 \leq \gamma \leq 1$). This is done to value rewards received earlier higher than those received later, reflecting the value of a good start. The algorithm therefore has the following Q-function that calculates the quality of a state-action combination, $\mathbf{Q} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbb{R}$. At the start of the algorithm, Q is initialized to an arbitrary fixed value. Then, at each time, t, the agent selects an action a_t, observes a reward r_t, enters a new state s_{t+1} and Q is updated. The value iteration updates are done using the dynamic programming based Bellman equation using the weighted average of the old value and the new information as follows:

$$\mathrm{Q}^{\mathrm{new}}(s_t, a_t) \leftarrow \mathrm{Q}(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot \max_a \mathrm{Q}(s_{t+1}, a_t) - \mathrm{Q}(s_t, a_t))$$

where α is the learning rate ($0 < \alpha \leq 1$). An episode of the algorithm ends when state, s_{t+1}, is a terminal state. In our tableau tree, open complete ABoxes are the terminal states. An example of the reward function may be to award a value of 1 after each action that doesn't result in either contradiction or in a terminal state while coming from top to down in the tree and award a value of 10 on reaching a terminal state while awarding a value of -10 on reaching a contradiction node and award -1 on each successive step during backtracking due to occurrence of a contradiction. Deep Q-Learning a variant of the above algorithm that uses convolutional deep neural networks to simulate the above framework.

When the Q-learning algorithm runs for sufficient number of episodes and learns the environment, then we can define the confidence scores of the non-deterministic choices at any node in the tree. These confidence scores represent the likelihood of getting success on a choice among a host of non-deterministic options.

We are currently implementing this approach and upon its completion, we plan to evaluate its performance on varying size OWL 2 DL ontologies. For this purpose, we would utilize our recently developed benchmark – OWL2Bench (Singh, Bhatia, and Mutharaju 2020).

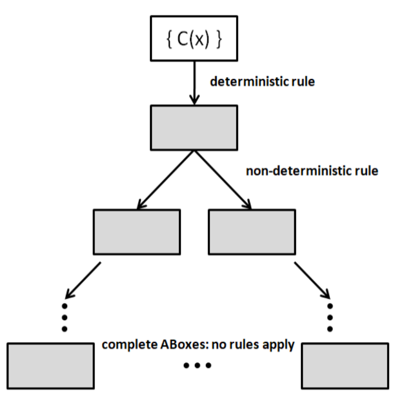

Figure 1: Tableau Search Tree

References

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the EL Envelope. LTCS-Report.

d' Amato, C. 2020. Machine Learning for the Semantic Web: Lessons Learnt and Next Research Directions. *Semantic Web Journal* 11: 1–9.

Hitzler, P.; Bianchi, F.; Ebrahimi, M.; and Sarker, M. K. 2019. Neural-symbolic integration and the Semantic Web. *Semantic Web Journal* 11: 1–9.

Kulmanov, M.; Liu-Wei, W.; Yan, Y.; and Hoehndorf, R. 2019. EL Embeddings: Geometric Gonstruction of Godels for the Description logic EL++. In *IJCAI 2019*, 6103–6109.

Möller, R.; and Haarslev, V. 2009. *Tableau-Based Reasoning*, 509–528.

Singh, G.; Bhatia, S.; and Mutharaju, R. 2020. OWL2Bench: A Benchmark for OWL 2 Reasoners. In *ISWC 2020*.