Markov Logic Networks for Natural Language Question Answering

Tushar Khot, Niranjan Balasubramanian, Eric Gribkoff, Ashish Sabharwal, Peter Clark, Oren Etzioni

QA via Logical Reasoning

Task: Answer 4th grade multiple choice science questions:

> *A fox grows thick fur as the season changes. This helps the fox to*
> *(A) hide from danger*
> *(B) attract a mate*
> *(C) find food*
> *(D) keep warm*

Approach:

1) **Create Knowledge Base** KB: Parse sentences in 4th grade texts into logical first-order rules, LHS => RHS

> *Growing thicker fur in winter helps some animals to stay warm*
> *isa(g, "grow"), isa(a, "some_animals"), isa(f, "thicker_fur"), isa(w, "the_winter"), agent(g, a), object(g, f), in(g, w) →*
> *∃ s, r: isa(s, "stays"), isa(r, "warm"), enables(g, s), agent(s, a), object(s, r)*

2) **Parse question** into a *Setup* portion that is asserted to be true and *Query* whose veracity is to be assessed:

> *Is it true that a fox grows thick fur to keep warm ?*
> *Setup : isa(F,"fox"), isa(G, "grows"), isa(T, "thick_fur"), agent(G, F), object(G, T)*
> *Query : isa(K, "keep_warm"), enables(G, K), agent(K, F)*

3) **Use logical reasoning** to prove (or find evidence for) the query, given the setup and KB

Key Challenges

- **Lexical variability, textual entailment**. E.g. "thick_fur" vs "thicker_fur"; "fox" vs "some_animals"
- **Text-derived rules are incomplete or over-specified**, making rule application in a pure logical setting brittle. E.g., naive application of the above rule wouldn't conclude the query as the rule requires "in the winter" to be true.
- **Rules may need to be chained** as a single text-derived rule may be insufficient to answer a question. E.g., chain "Animals grow thick fur in winter" and "Thick fur helps keep warm".

Three MLN Formulations

A. First Order MLN
- Pros: A natural formulation, uses KB rules essentially directly as first-order MLN clauses
- Cons: Struggles with long conjunctions + existentials, relatively few atoms, little to no symmetries
  - Benefits from exploiting structure imposed by hard constraints to vastly simplify groundings

B. Entity Resolution MLN

Express generalities over classes of individuals by replacing first-order vars with prototypical constants
- Pros: Reduces the number of groundings, while retaining the crux of the reasoning problem
- Cons: Too brittle in handling lexical mismatches

C. Praline (PRobabilistic ALignment and INferencE)

Inference using primarily the string constants but guided by predicate alignment.
- Pros: Relaxes rigidity in rule application by explicitly modeling the desired QA inference behavior
- 15% accuracy boost, 10x reduction in runtime
- Cons: Introduces additional complexity to define and control inference

(A) First Order MLN

- Straightforward KB translation is extremely inefficient
- We refine the formulation in two ways:
  - **Refined types** - entities, events, and strings; predicates are appropriately typed
  - **Semantic Rules** - capture the intended meaning of our predicates, e.g. every event has a unique agent
- External alignment function (based on WordNet) is used to estimate entailment between words, and from setup to antecedent and consequent to query
- Address existentials spanning conjunction in KB by introducing a new existential predicate

Efficient Inference

- Our QA encodings have **small domain sizes** and hence **very few ground atoms**
  - Most existing lifted & lazy inference techniques are inspired by large number of ground atoms, and were ineffective for our models.
  - Lazy inference: reduced 70K ground clauses to 56K
  - LBG and PTP (Alchemy-2) : slower than Alchemy-1
- Our approach to reducing grounding size:
  1. Generate propositional grounding of 1 MLN clause
  2. **Use a propositional SAT solver** to identify the Backbone variables G (subsumes Unit Propagation)
  3. Freeze the values of G; repeat this process
  4. Remove all frozen variables in the end
- *Our method brought 70K ground clauses down to only 951 clauses in the above example*

(B) Entity Resolution MLN

- Representing generalities as quantified rules appears to be a natural formulation, but is also quite inefficient
- Idea: instead treat generalities as relations expressed over **prototypical entities and events**, inspired by existing literature on Entity Resolution with MLNs
- A first-order rule in FO MLN is now fully grounded:

> *isa(G, "grow"), isa(A, "some animals"), isa(F, "thicker fur"), isa(W, "the winter"), agent(G, A), object(G, F), in(G, W) → isa(S, "stays"), isa(R,"warm"), enables(G, S), agent(S, A), object(S, R)*

- Defines soft clusters or equivalence classes of entities and events, through a probabilistic ***sameAs*** predicate which is reflexive, symmetric, and transitive:

$$w(s, s') : \; entails(s, s')$$
$$isa(x, s), entails(s, s') \rightarrow isa(x, s').$$
$$isa(x, s), isa(y, s) \rightarrow sameAs(x, y).$$
$$w : \; isa(x, s), !isa(y, s) \rightarrow !sameAs(x, y)$$
$$r(x, y), sameAs(y, z) \rightarrow r(x, z).$$

Experiments

- **Benchmark**: Elementary-level science questions (non-diagram, multiple-choice) from 4th grade New York Regents exam
  - Dev set: 108 questions
  - Test set: 68 questions (unseen)
- **KB** generated in advance by processing the 4th grade science exam syllabus, Barron's study guide, and querying the Internet for relevant terms (~47,000 rules)

Question Set	MLN Formulation	#Answered (some / all)	Exam Score	#MLN Rules	#Atoms	#Ground Clauses	Runtime (all)
Dev-108	FO-MLN	106 / 82	33.6%	35	384*	524*	280 s
	ER-MLN	107 / 107	34.5%	41	284	2,308	188 s
	PRALINE	108	**48.8%**	51	182	219	**17 s**
Unseen-68	FO-MLN	66	33.8%	-	-	-	288 s
	ER-MLN	68	31.3%	-	-	-	226 s
	PRALINE	68	**46.3%**	-	-	-	**17 s**

(C) Praline [PRobabilistic ALignment and INferencE]

- Approaches (A) and (B) are **still rigid** in two aspects:
  - Even if the *words* match exactly, the rule will still not "fire" if the *edges* or relations do not match
  - Clustering forces entities bound to lexically equivalent strings to "behave" identically, but questions may contain two different entities bound to equivalent string representations
- **Praline defines flexible model** to additionally handle the above shortcomings as well as:
  - Acyclic inference: QA must avoid feedback loops
  - False Unless Proven**:** Atoms are false unless stated in the question or proven through the application of a rule.
- Defines a unary predicate, ***holds***, over string constants to capture what is known to be true or can be proven to be true (via inference) in the world

  holds(Grow), holds(Animals), holds(Fur), holds(Winter) →
  holds(Stays), holds(Warm)

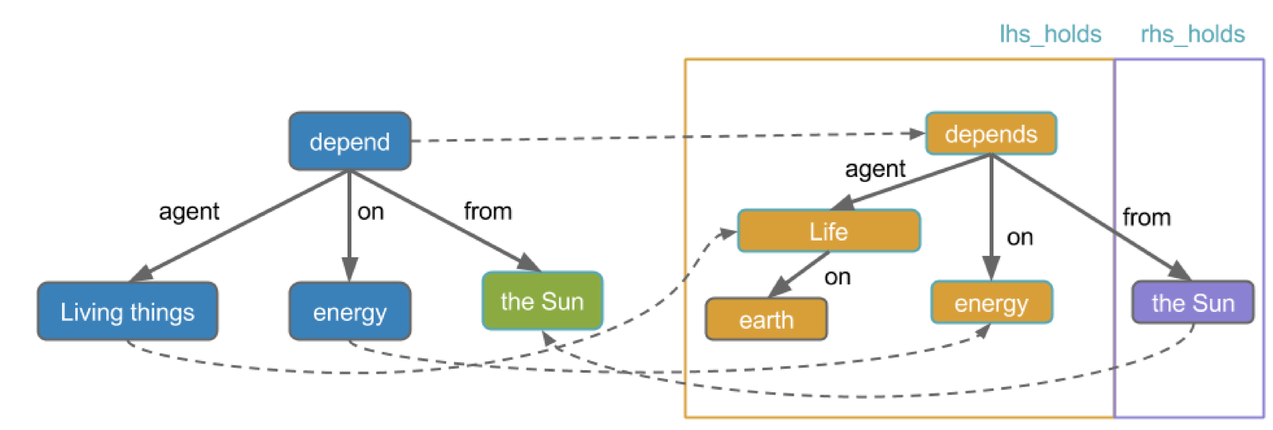
- **Graph alignment rules** use entailment & nbrhood info:

  aligns(x, y), edge(x, u, r), edge(y, v, s) → aligns(u, v)

- **Inference rules** define what can be concluded to be true given the setup, either based on alignment or rule application

  holds(x), aligns(x, y) → holds(y)
  lhsHolds(r) → rhsHolds(r)

- **Acyclic inference**
  - Predicates ***proves*** and ***ruleProves*** to capture the inference chain
  - Ensure acyclicity in inference by introducing transitive clauses and disallowing reflexivity
- **False Unless Proven**
  - Add bidirectional implications on all clauses
  - Alternative: introduce a strong negative prior; however predictions were too sensitive to the negative prior



Conclusion & Future Work

- Investigated potential of MLNs for QA resulting in multiple formulations; **Praline provided a flexible model that outperformed more natural approaches**.
- While SRL seems a perfect fit, **simpler word-overlap based approaches** were better on this dataset (~55%)
  - Increased flexibility of complex relational models comes at the cost of increased susceptibility to noise in the input. Automatically learning weights of these models may better leverage this flexibility.
- Modeling the QA task with an **undirected model** gives the flexibility to define a joint model that allows alignment to influence inference and vice versa.
  - However, inference chains themselves are acyclic, suggesting models such as Problog and SLP may be a better fit for this sub-task.

Dev set and MLNs available at:
http://allenai.org/software.html

Any questions ? Contact us at:
tushark/ashishs@allenai.org

ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE
Stony Brook University
UNIVERSITY of WASHINGTON