Exploring Markov Logic Networks for Question Answering

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

ACL, 2015, AllenNLP/University of Washington

Leon Zhou (zhliyang@seas.upenn.edu)

4/16/2020

Penn Engineering

The Great AI Schism

Field	Statistical Approach	Logical Approach
Knowledge Representation	Graphical Models	First Order Logic
Automated Reasoning	Statisfiability Testing	Markov Chain
Machine Learning	Inductive Logic Programming	Neural Nets
Planning	Classical Planning	MDP
NLP	Definite Clause Grammer	Prob. Context Free Grammar

Problem & Motivation

- Elementary-Science Exam QA
  - Challenges in knowledge acquisition & reasoning
  - Automatically extracted knowledge (scalable, noisy, incomplete)
  - Reasoning mechanism to handle uncertainty

- E.g.

Knowledge: gravity pulls objects towards the Earth

Question: which force is responsible for a ball to drop?

Problem & Motivation

- Input (k multiple choices as T/F)
  - Knowledge base / Rules **KB** (textual resources)
  - Setup **S** (known facts)
  - Question Choices **Q** (k choices)

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

- Output (most likely answer as inference)

$$\arg\max_{i \in \{1,\ldots,k\}} \Pr[Q_i \mid S, KB]$$

Progress to Date:

- Probabilistic logic [Nilsson, 1986]
- Statistics and beliefs [Halpern, 1990]
- Knowledge-based model construction [Wellman et al., 1992]
- Stochastic logic programs [Muggleton, 1996]
- Probabilistic relational models [Friedman et al., 1999]
- Relational Markov networks [Taskar et al., 2002]
- …
- this paper: Markov Logic Network

Contents:

- **First Order Logic**
- Markov Logic Network
- Probabilistic Formulations
- First-Order MLN (attempt 1)
- Entity Resolution MLN (attempt 2)
- Praline MLN (best attempt)
- Results

First Order Logic

- Constants, variables, functions, predicates

E.g. : Anna, x, MotherOf(x), Friends(x,y)

- Grounding: Replace all variables by constants

E.g. : Friends (Anna, Bob)

- World (model, interpretation):

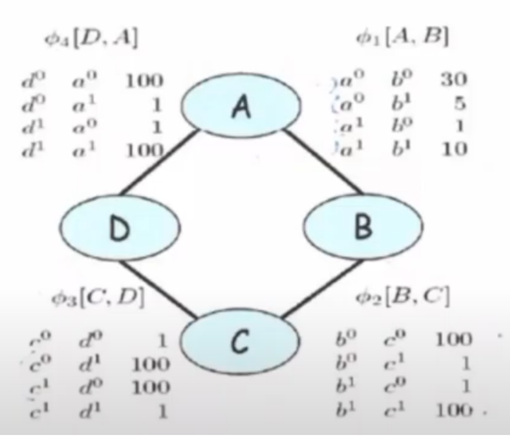
Assignment of truth values to all ground predicates

$$\forall x \; Smokes(x) \Rightarrow Cancer(x)$$
$$\forall x, y \; Friends(x, y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$$

Markov Network

- Undirected graphical models

- Cliques with weights/potential functions



$$P(x)=\frac{1}{Z}\exp\left(\sum_i \underbrace{w_i}_{\text{Weight of Feature } i}\,\underbrace{f_i(x)}_{\text{Feature } i}\right)$$

Markov Logic

- Syntax: Weighted first-order formulas

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$

- Semantics: Templates for Markov network

Two constants: **Anna** (A) and **Bob** (B)

1.5 Smokes(A) => Cancer(A)
1.5 Smokes(B) => Cancer(B)
1.1 Friends(A,A) => (Smokes(A) <=> Smokes(A)
1.1 Friends(B,B) => (Smokes(B) <=> Smokes(B))
1.1 Friends(A,B) => (Smokes(A) <=> Smokes(B))
1.1 Friends(B,A) => (Smokes(B) <=> Smokes(A))

$W(I) = \exp(\sum_{\substack{w:F \\ I \models F}} w)$

I = {Friends(A,A), Friends(A,B), Friends(B,A), Friend(B,B), Smokes(B)}
P(I) =

Markov Logic

- Syntax: Weighted first-order formulas

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$

- Semantics: Templates for Markov network

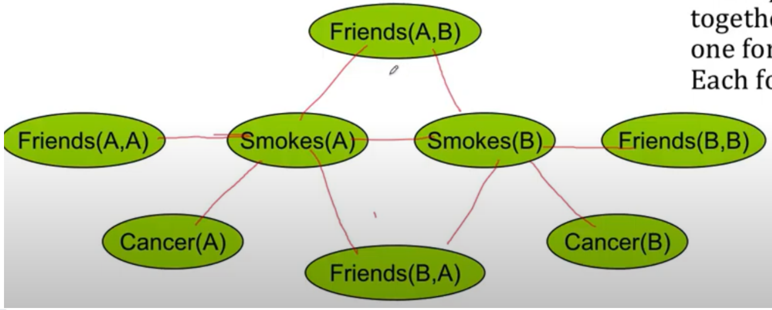
Markov Logic

- Syntax: Weighted first-order formulas

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$

- Semantics: Templates for Markov network

Two constants: **Anna** (A) and **Bob** (B)



Edge between two nodes iff the corresponding ground atoms appear together in at least one grounding of one formula.
Each formula induces a clique

Markov Logic Network

- Syntax: Weighted first-order formulas
- Semantics: Templates for Markov nets
- Example

Knowledge: gravity pulls objects towards the Earth

Question: which force is responsible for a ball to drop?

Contents:

- First Order Logic
- Markov Logic Network
- **Probabilistic Formulations**
- First-Order MLN (attempt 1)
- Entity Resolution MLN (attempt 2)
- Praline MLN (best attempt)
- Results

Probablistic Formulations

- **Rule Representation:**
- "Growing thicker fur in winter helps some animals to stay warm"

$$\begin{aligned}&isa(g, \text{grow}), isa(a, \text{some animals}),\\&isa(f, \text{thicker fur}), isa(w, \text{the winter}),\\&agent(g, a), object(g, f), in(g, w)\end{aligned} \quad \begin{aligned}\Rightarrow\ &\exists s, r : isa(s, \text{stays}), isa(r, \text{warm}),\\&enables(g, s), agent(s, a), object(s, r)\end{aligned}$$

- **Question Representation:**
- Setup: A fox grows thick fur as the season changes.
- Choices: This helps the fox to (A) hide from danger (B) attract a mate (C) find food (D) keep warm?

$$\begin{aligned}setup :&isa(F, \text{fox}), isa(G, \text{grows}), isa(T,\\&\text{thick fur}), agent(G, F), object(G, T)\\query :&isa(K, \text{keep warm}), enables(G, K),\\&agent(K, F)\end{aligned}$$

Penn Engineering

$$\arg\max_{i \in \{1,\ldots,k\}} \Pr[Q_i \mid S, KB]$$

Contents:

- First Order Logic
- Markov Logic Network
- Probabilistic Formulations
- **First-Order MLN (attempt 1)**
- Entity Resolution MLN (attempt 2)
- Praline MLN (best attempt)
- Results

First Order MLN

- QA task of Pr[Qi| S, R] as an MLN program M
  - add R essentially verbatim as first-order rules in M
  - Predicates of M: ones in R and entails predicate : "thick fur" & "thicker fur" "fox" & "some animals"
  - Evidence:
  - Soft evidence for M consists of entails relations between every ordered pair of entity (or event). Hard evidence for M comprises of grounded atoms in S.
  - Query: The query atom in M is result(). We are interested in computing Pr[result() = true].
  - Semantic Rules: rules that capture the intended meaning of our predicates, such as every event has a unique agent, cause(x, y) →effect(y, x)
  - DrawBack:Computationally ineffcient, large grounded network

Entity Resolution MLN

- Prototypical entity/event constants
  - String constants instead of first order variables

$$agent(Grow, Animals), object(Grow, Fur) \Rightarrow enables(Grow, StayWarm)$$

  - Previously

$$\begin{aligned}
& isa(g, \text{grow}), isa(a, \text{some animals}), \\
& isa(f, \text{thicker fur}), isa(w, \text{the winter}), \\
& agent(g, a), object(g, f), in(g, w) \\
& \Rightarrow \exists s, r : isa(s, \text{stays}), isa(r, \text{warm}), \\
& \quad enables(g, s), agent(s, a), object(s, r)
\end{aligned}$$

  - Equivalence or Resolution Rules: sameAs predicate

$$isa(x, s), entails(s, s') \rightarrow isa(x, s'). \qquad r(x, y), sameAs(y, z) \rightarrow r(x, z).$$

$$isa(x, s), isa(y, s) \rightarrow sameAs(x, y).$$

$$w: \ isa(x, s), !isa(y, s) \rightarrow !sameAs(x, y)$$

Entity Resolution MLN

- Partial Match Rules:

$$(\wedge_{i=1}^{k} L_i) \rightarrow R \quad \longrightarrow \quad L_i \rightarrow R$$

- Drawbacks: the entailment-based clusters of constants always behave similarly
- fail on questions that have distinct entities with similar string representations

(e.g. two distinct plants in a question would map to the same entity).

- fails to apply valid rules in the presence of syntactic differences

agent(Fall, Things) generated by "things fall due to gravity" and object(Dropped, Ball) for "a student dropped a ball".

PRobabilistic ALignment and INferencE

- Controlled Inference Given KB
- Acyclic Inference, False Unless Proven
- predicate **holds** : a unary predicate over string constatns

$$\begin{aligned}&isa(g, \text{grow}), isa(a, \text{some animals}),\\ &isa(f, \text{thicker fur}), isa(w, \text{the winter}),\\ &agent(g, a), object(g, f), in(g, w)\\ &\Rightarrow \exists s, r : isa(s, \text{stays}), isa(r, \text{warm}),\\ &\quad enables(g, s), agent(s, a), object(s, r)\end{aligned} \longrightarrow \begin{aligned}&holds(Grow), holds(Animals), holds(Fur),\\ &holds(Winter) \Rightarrow holds(Stays), holds(Warm)\end{aligned}$$

PRobabilistic ALignment and INferencE

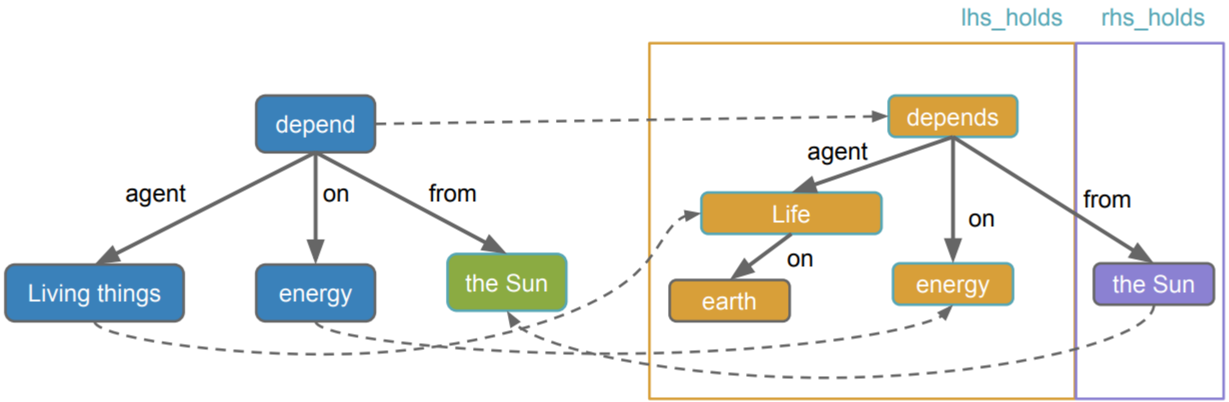
- Graph Alignment Rules:

$$aligns(x, y), edge(x, u, r), edge(y, v, s) \Rightarrow aligns(u, v)$$

- Inference Rules

$$holds(x), aligns(x, y) \Rightarrow holds(y)$$

PRobabilistic ALignment and INferencE



blue: setup;
green:query;
orange:antecedent;
purple:consequent;
dotted lines: alignments.
lhsHolds combines individual probabilities of antecedent nodes and
rhsHolds captures the probability of the consequent.

Penn Engineering

Contents:

- First Order Logic
- Markov Logic Network
- Probabilistic Formulations
- First-Order MLN (attempt 1)
- Entity Resolution MLN (attempt 2)
- Praline MLN (best attempt)
- **Contributions & Results**

Results and Analysis

- Contributions:
- KB (roughly 47,000 sentences)
- Dataset (non-diagram, multiple-choice)
- MLN models * 3

Results and Analysis

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**

	Dev-108	Unseen-68	Dev-170	Unseen-176
Praline	50.3%	**52.7%**	33.2%	36.6%
Word-based	**57.4%**	51.5%	**40.3%**	**43.3%**

Conclusions, Shortcomings and Future Work

- Reasoning with automatically extracted knowledge:
  - very hard
  - first-order representations are highly inefficient
  - structural variability makes it harder
  - not up to par with textual feature-based approaches (Beltagy and Mooney, 2014)
- Potential fix:

  Automatically learning weights might leverage model flexibility

Questions ?