Natural Language Semantics using Probabilistic Logic

Islam Beltagy

Doctoral Dissertation Defense

Supervising Professors: Raymond J. Mooney, Katrin Erk

Who is the first president of the United States ?

– George Washington
– "*George Washington was the first President of the United States, the Commander-in-Chief of the Continental Army and one of the Founding Fathers of the United States*"

Where was George Washington born ?

– Westmoreland County, Virginia
– "*George Washington was born at his father's plantation on Pope's Creek in Westmoreland County, Virginia*"

What is the birthplace of the first president of the United States ?

– …. ???

Objective

Develop a new **semantic representation**

With better semantic representations, more NLP applications can be done better

- Automated Grading, Machine Translation, Summarization, Question Answering …

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
- Future work
- Conclusion

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
- Future work
- Conclusion

Formal Semantics

Natural language ➜ Formal language

[Montague, 1970]

A person is driving a car

$$\exists x,y,z.\ \mathrm{person}(x) \wedge \mathrm{agent}(y,x) \wedge \mathrm{drive}(y) \wedge \mathrm{patient}(y,z) \wedge \mathrm{car}(z)$$

✅ Expressive: entities, events, relations, negations, disjunctions, quantifiers …

✅ Automated inference: theorem proving

❌ Brittle: unable to handle uncertain knowledge

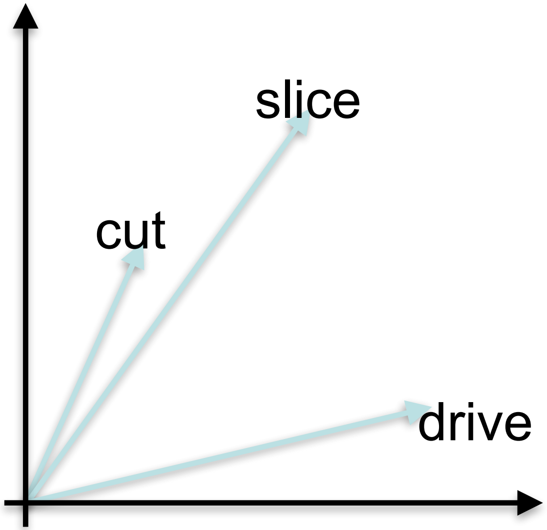
Distributional Semantics

"You shall know a word by the company it keeps" [John Firth, 1957]

Word as vectors in high dimensional space

✅ Captures graded similarity

❌ Does not capture structure of the sentence

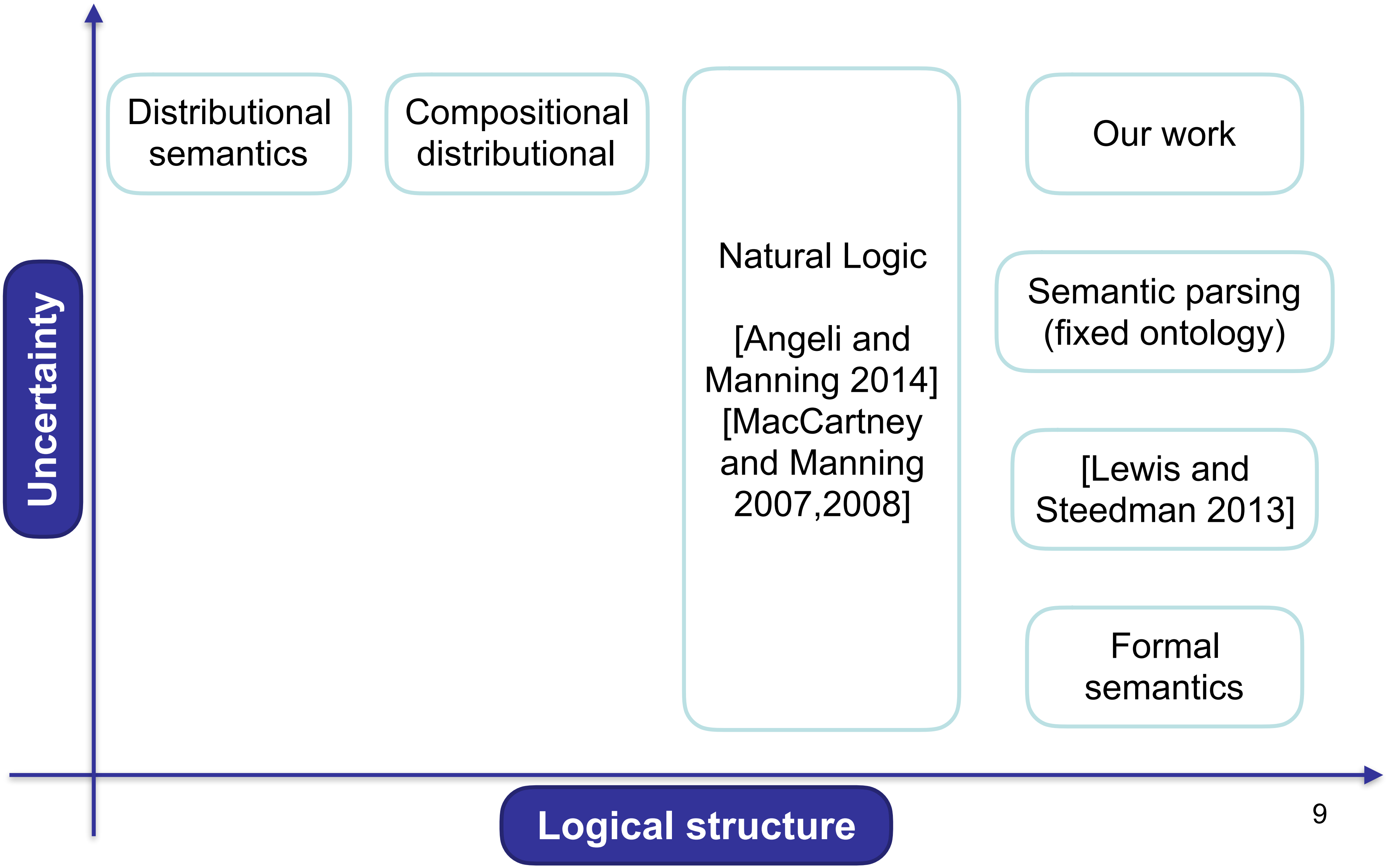


Proposal: Probabilistic Logic Semantics
[Beltagy et al., *SEM 2013]

Probabilistic Logic

- Logic: expressivity of formal semantics
- Reasoning with uncertainty:
  - encode linguistic resources
    - e.g: distributional semantics

Related Work



Proposal: Probabilistic Logic Semantics

Logic + Statistics [Nilsson, 1986][Getoor and Taskar, 2007]

Weighted first-order logic rules	$\forall x.\ \mathrm{slice}(x) \rightarrow \mathrm{cut}(x) \mid 2.3$ $\forall x.\ \mathrm{apple}(x) \rightarrow \mathrm{company}(x) \mid 1.6$

Implementations

- Markov Logic Networks (MLNs) [Richardson and Domingos, 2006]
- Probabilistic Soft Logic (PSL) [Kimmig et al., NIPS 2012]

Proposal: Probabilistic Logic Semantics

Logic + Statistics [Nilsson, 1986][Getoor and Taskar, 2007]

Weighted first-order logic rules

$$\forall x.\ \mathrm{slice}(x) \rightarrow \mathrm{cut}(x) \quad | \ 2.3$$

$$\forall x.\ \mathrm{apple}(x) \rightarrow \mathrm{company}(x) \quad | \ 1.6$$

Distributional similarity

WSD confidence

Implementations

- Markov Logic Networks (MLNs) [Richardson and Domingos, 2006]
- Probabilistic Soft Logic (PSL) [Kimmig et al., NIPS 2012]

Markov Logic Networks

[Richardson and Domingos, 2006]

Weighted first-order logic rules

$$\forall x, y.\ \mathrm{ogre}(x) \wedge \mathrm{friend}(x, y) \rightarrow \mathrm{ogre}(y) \quad |\ 1.1$$

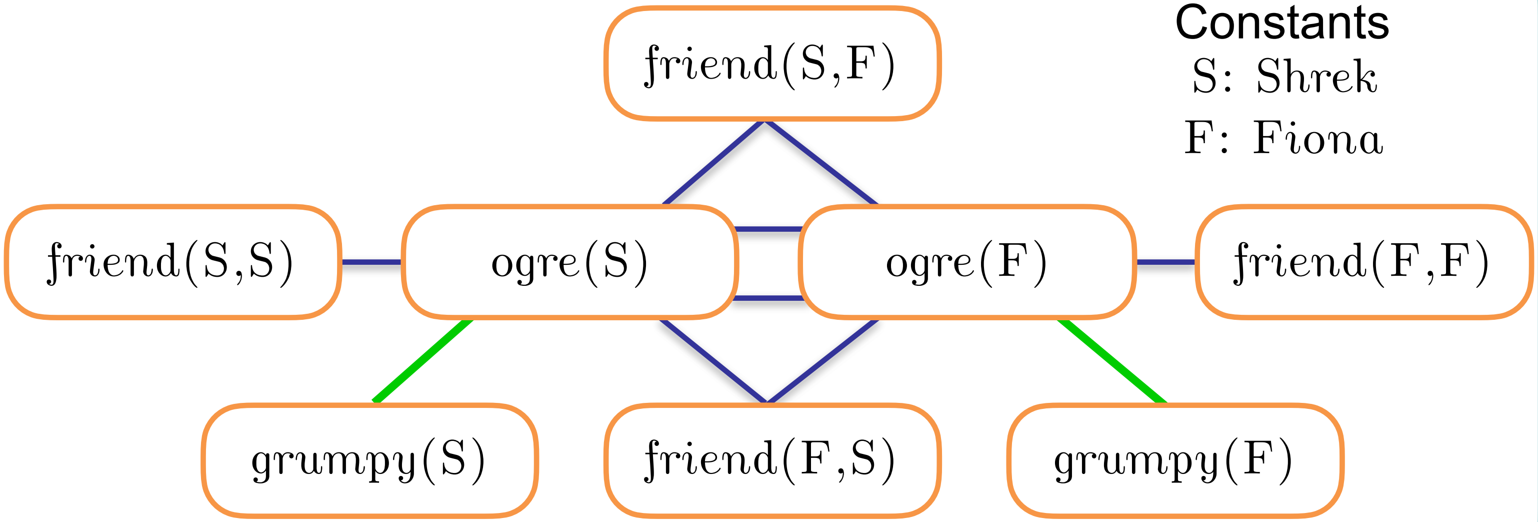
$$\forall x.\ \quad \mathrm{ogre}(x) \rightarrow \mathrm{grumpy}(x) \quad |\ 1.5$$

Graphical model: Probability distribution over possible worlds

Constants

S: Shrek

F: Fiona



Inference $P(Q|E,KB)$

$$P(\mathrm{grumpy}(\mathrm{Shrek})\ |\ \mathrm{friend}(\mathrm{Shrek}, \mathrm{Fiona}), \mathrm{ogre}(\mathrm{Fiona}))$$

Markov Logic Networks

[Richardson and Domingos, 2006]

Probability Mass Function (PMF)

$$P(x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right)$$

a possible truth assignment

Normalization constant

Weight of formula i

No. of true groundings of formula i in x

PSL: Probabilistic Soft Logic

[Kimmig et al., NIPS 2012]

Designed with focus on efficient inference

Atoms have continuous truth values $\in$ [0,1] (MLN: Boolean atoms)

Łukasiewicz relaxation of AND, OR, NOT

- $\mathrm{I}(\ell 1 \wedge \ell 2) = max\ \{0,\ \mathrm{I}(\ell 1) + \mathrm{I}(\ell 2) - 1\}$
- $\mathrm{I}(\ell 1 \vee \ell 2) = min\ \{1,\ \mathrm{I}(\ell 1) + \mathrm{I}(\ell 2)\ \}$
- $\mathrm{I}(\neg\ \ell 1) \qquad = 1 - \mathrm{I}(\ell 1)$

Inference: linear program (MLN: combinatorial counting problem)

PSL: Probabilistic Soft Logic

[Kimmig et al., NIPS 2012]

PDF: $$f(I) = \frac{1}{Z}\exp[-\sum_{r \in R} \lambda_r (d_r(I)) \;]$$

a possible continuous truth assignment

Normalization constant

For all rules

Weight of formula r

Distance to satisfaction of rule r

Inference: Most Probable Explanation (MPE)

– Linear program

Tasks

Require deep semantic understanding

- Textual Entailment (RTE) [Beltagy et al., 2013,2015,2016]
- Textual Similarity (STS) [Beltagy et al., 2014] (proposal work)
- Question Answering (QA)

Pipeline for an Entailment

- T: A person is driving a car
- H: A person is driving a vehicle

Does $\mathrm{T} \models \mathrm{H}$?

Logical form

- $\exists x, y, z.\ \ \mathrm{person}(x) \wedge \mathrm{agent}(y,\ x) \wedge \mathrm{drive}(y) \wedge \mathrm{patient}(y,\ z) \wedge \mathrm{car}(z)$
- $\exists x, y, z.\ \ \mathrm{person}(x) \wedge \mathrm{agent}(y,\ x) \wedge \mathrm{drive}(y) \wedge \mathrm{patient}(y,\ z) \wedge \mathrm{vehicle}(z)$

Knowledge base

- KB: $\forall x.\ \mathrm{car}(x) \rightarrow \mathrm{vehicle}(x) \mid w$

Inference

- Calculating $\mathrm{P}(\mathrm{H} | \mathrm{T},\ \mathrm{KB})$

Summery of proposal work

- Efficient MLN inference for the RTE task [Beltagy et al., 2014]
- MLNs and PSL inference for the STS task [Beltagy et al., 2013]
- Reasons why MLNs fit RTE and PSL fits STS

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
- Future work
- Conclusion

Logical form

- T: A person is driving a car
- H: A person is driving a vehicle

Parsing

- T: $\exists x,y,z.\ \ \mathrm{person}(x) \wedge \mathrm{agent}(y, x) \wedge \mathrm{drive}(y) \wedge \mathrm{patient}(y, z) \wedge \mathrm{car}(z)$
- H: $\exists x,y,z.\ \ \mathrm{person}(x) \wedge \mathrm{agent}(y, x) \wedge \mathrm{drive}(y) \wedge \mathrm{patient}(y, z) \wedge \mathrm{vehicle}(z)$

Using **Boxer**, a rule based system on top of a CCG parser

[Bos, 2008]

Adapting logical form

Theorem proving: $\mathrm{T} \wedge \mathrm{KB} \models \mathrm{H}$

Probabilistic logic: $\mathrm{P}(\mathrm{H}|\mathrm{T},\mathrm{KB})$

- Finite domain: explicitly introduce needed constants
- Prior probabilities: results are sensitive to prior probabilities

Adapt logical form to probabilistic logic

Adapting logical form

[Beltagy and Erk, IWCS 2015]

Finite domain (proposal work)

– **Quantifiers** don’t work properly

T: Tweety is a bird. Tweety flies $\quad \mathrm{bird}(\text{🐤}) \wedge \mathrm{agent}(F, \text{🐤}) \wedge \mathrm{fly}(F)$

H: All birds fly $\quad \forall x.\ \mathrm{bird}(x) \rightarrow \exists y.\ \mathrm{agent}(y, x) \wedge \mathrm{fly}(y)$

Solution: additional entities

Add an extra
$\mathrm{bird}(\text{🐧})$

Adapting logical form

[Beltagy and Erk, IWCS 2015]

Prior probabilities

- Ground atoms have prior probability 0.5
- $\mathrm{P(H|KB)}$ determines how useful $\mathrm{P(H|T,KB)}$ is
- If both values are high
  - T entails H
  - Prior probability of H is high
- Example
  - T: My car is green
  - H: There is a bird
- Goal: Make $\mathrm{P(H|T,KB)}$ less sensitive to $\mathrm{P(H|KB)}$

Adapting logical form
[Beltagy and Erk, IWCS 2015]

Prior probabilities

– Solution 1: use the ratio $\dfrac{P(H \mid T, KB)}{P(H \mid KB)}$

– Not a good fit for the Entailment task
  - T: A person is driving a car
  - H: A person is driving a green car
  - The ratio is high but $T \not\models H$

Adapting logical form

[Beltagy and Erk, IWCS 2015]

Prior probabilities

- Solution 2: set ground atom priors such that $\mathrm{P}(\mathrm{H}|\mathrm{KB}) \approx 0$
- Matches the definition of the Entailment task
  - T: Obama is the president of the USA
  - H: Austin is in Texas
  - Even though H is true in the real world, $T \not\models H$

Adapting logical form
[Beltagy and Erk, IWCS 2015]

Prior probabilities

- Solution 2: set ground atom priors such that $\mathrm{P}(\mathrm{H}|\mathrm{KB}) \approx 0$
  - Ground atoms not entailed by $\mathrm{T} \wedge \mathrm{KB}$ are set to false
    - (everything is false by default)
  - Prior probability of <u>negated predicates of H</u> is set to high value
    - T: A dog is eating
    - H: A dog does <u>not fly</u>

Adapting logical form

[Beltagy and Erk, IWCS 2015]

Evaluation — Entailment datasets

some, all,
no, not all

all monotonicity
directions

- Synthetic
  - T: No man eats all delicious food
  - H: Some hungry men eat not all food

Adapting logical form
[Beltagy and Erk, IWCS 2015]

Evaluation — Entailment datasets

- SICK [SemEval 2014] (5K training, 5K testing)
  - Short video description sentences
  - Example
    - » T: A young girl is dancing
    - » H: A young girl is standing on one leg
- FraCas [Cooper et al., 1996]
  - 46 manually constructed entailments to evaluate quantifiers
  - Example:
    - » T: A Swede won a Nobel prize. Every Swede is a Scandinavian
    - » H: A Scandinavian win a Nobel prize

Adapting logical form

[Beltagy and Erk, IWCS 2015]

Evaluation — Results

	Synthetic	SICK	FraCas
No adaptations	50.78%	68.10%	50.00%
Finite domain	82.42%	68.14%	63.04%
Finite domain + priors	100%	76.52%	100.0%

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
- Future work
- Conclusion

Knowledge Base

Logic handles sentence structure and quantifier

+

Knowledge base encodes lexical information

Knowledge Base

[Beltagy et al., CompLing 2016]

Collect the relevant **<u>weighted</u>** KB from different resources

Precompiled rules

- WordNet rules: map semantic relations to logical rules
- Paraphrase rules: translate PPDB to weighted logical rules

Generate on-the-fly rules for a specific dataset/task

- Lexical resources are never complete

On-the-fly rules

[Beltagy et al., CompLing 2016]

Simple solution: (proposal work)

- Generate rules between all pairs of words
- Use distributional similarity to evaluate the rules

T: A person is driving a car

H: A person is driving a vehicle

- Generating a lot of useless rules
- Generated rules have limited predefined forms

On-the-fly rules
[Beltagy et al., CompLing 2016]

Better solution:

– Use the logic to <u>propose</u> relevant lexical rules
– Use the training set to learn rule weights

On-the-fly rules
[Beltagy et al., CompLing 2016]

1) Rules proposal: using Robinson resolution

T: $\mathrm{person}(P) \wedge \mathrm{agent}(D, P) \wedge \mathrm{drive}(D) \wedge \mathrm{patient}(D, C) \wedge \mathrm{car}(C)$

H: $\exists x,y,z.\ \mathrm{person}(x) \wedge \mathrm{agent}(y, x) \wedge \mathrm{drive}(y) \wedge \mathrm{patient}(y, z) \wedge \mathrm{vehicle}(z)$

T: $\mathrm{car}(C)$

H: $\mathrm{vehicle}(C)$

Proposed rules:

KB: $\forall x.\ \mathrm{car}(x) \rightarrow \mathrm{vehicle}(x)$

On-the-fly rules

[Beltagy et al., CompLing 2016]

Example: complex rule

T: A person is solving a problem

H: A person is finding a solution to a problem

KB: $\forall e,x.\ solve(e) \wedge patient(e,x) \rightarrow \exists s.\ find(e) \wedge patient(e,s) \wedge solution(s) \wedge to(t,x)$

On-the-fly rules

[Beltagy et al., CompLing 2016]

Example: negative rule

T: A person is driving

H: A person is walking

KB: $\forall x.\ drive(x) \rightarrow \ walk(x)$

On-the-fly rules

[Beltagy et al., CompLing 2016]

Automatically annotating rules

- proposed rules of
  - entailing examples: positive rules
  - non-entailing examples: negative rules

On-the-fly rules

[Beltagy et al., CompLing 2016]

- T: A man is walking $\models$ H: A person is walking
  - $\forall x.\ \mathit{man}(x) \rightarrow \mathit{person}(x)$ positive rule
- T: I have a green car $\not\models$ H: I have a green bike
  - $\forall x.\ \mathit{car}(x) \rightarrow \mathit{bike}(x)$ negative rule

On-the-fly rules

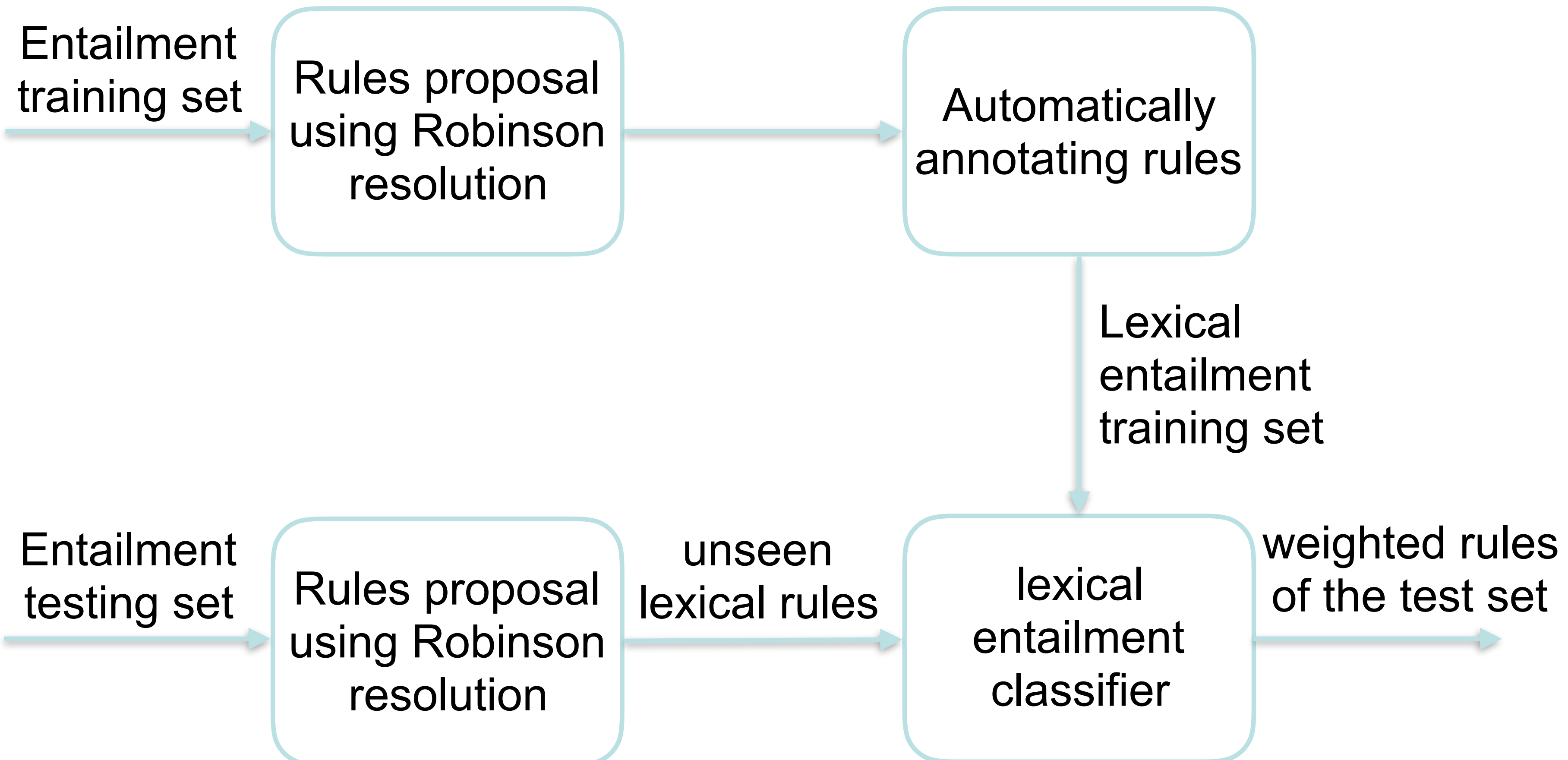
[Beltagy et al., CompLing 2016]

2) Weight learning

- The task of evaluating the lexical rules is called “lexical entailment”
- Usually viewed as a classification task (positive/negative rules)
  - We use the “lexical entailment classifier” by Roller and Cheng [Beltagy et al., CompLing 2016]
  - It uses various linguistic features to learn how to evaluate unseen rules
- Use the annotated rules of the training set to train the classifier
- Use the classifier to evaluate the rules of the test set
- Use classifier confidence as a rule weight

On-the-fly rules

[Beltagy et al., CompLing 2016]



On-the-fly rules
[Beltagy et al., CompLing 2016]

Entailment = Lexical Entailment + Probabilistic Logic Inference

On-the-fly rules — Evaluation

[Beltagy et al., CompLing 2016]

Recognizing Textual Entailment (RTE) [Dagan et al., 2013]

– Given two sentences T and H

– Find if T Entails, Contradicts or not related (Neutral) to H

Examples

– Entailment: T: A man is walking through the woods.
H: A man is walking through a wooded area.

– Contradiction: T: A man is jumping into an empty pool.
H: The man is jumping into a full pool.

– Neutral: T: A young girl is dancing.
H: A young girl is standing on one leg.

Textual Entailment — Settings

Logical form

- CCG parser + Boxer + Multiple parses
- Logical form adaptations
- Special entity coreference assumption for the detection of contradictions

Knowledge base

- Precompiled rules: WordNet + PPDB
- On-the-fly rules using Robinson resolution alignment

Inference

- $\mathrm{P}(\mathrm{H}|\mathrm{T},\ \mathrm{KB}),\ \mathrm{P}(\neg\mathrm{H}|\mathrm{T},\ \mathrm{KB})$
- Efficient MLN inference for RTE (proposal work)
- Simple rule weights mapping from [0-1] to MLN weights

Efficient MLN Inference for RTE

Inference problem: $\mathrm{P(H|T,\ KB)}$

Speeding up inference

Calculate probability of a complex query formula

Speeding up Inference

[Beltagy and Mooney, StarAI 2014]

MLN's grounding generates very large graphical models, especially in NLP applications

H has $\mathrm{O}(\mathrm{c}^{v})$ ground clauses

- v: number of variables in H
- c: number of constants in the domain

Speeding up Inference

[Beltagy and Mooney, StarAI 2014]

H: $\exists x, y.\ guy(x) \wedge agent(y, x) \wedge drive(y)$

Constants $\{A, B, C\}$

Ground clauses

$guy(A) \wedge agent(A, A) \wedge drive(A)$

$guy(A) \wedge agent(B, A) \wedge drive(B)$

$guy(A) \wedge agent(C, A) \wedge drive(C)$

$guy(B) \wedge agent(A, B) \wedge drive(A)$

$guy(B) \wedge agent(B, B) \wedge drive(B)$

$guy(B) \wedge agent(C, B) \wedge drive(C)$

$guy(C) \wedge agent(A, C) \wedge drive(A)$

$guy(C) \wedge agent(B, C) \wedge drive(B)$

$guy(C) \wedge agent(C, C) \wedge drive(C)$

Speeding up Inference

[Beltagy and Mooney, StarAI 2014]

Closed-world assumption: assume everything is false by default

- In the world, most things are false

Enables inference speeding up

- Large number of ground atoms are trivially false
- Removing them simplifies the inference problem
- Find these ground atoms using “evidence propagation”

Speeding up Inference

[Beltagy and Mooney, StarAI 2014]

T: $\mathrm{man}(M) \wedge \mathrm{agent}(D, M) \wedge \mathrm{drive}(D)$

KB: $\forall x.\ \mathrm{man}(x) \rightarrow \mathrm{guy}(x) \mid 1.8$

Ground Atoms:

$\mathrm{man}(M)$, $\mathrm{man}(D)$, $\mathrm{guy}(M)$, $\mathrm{guy}(D)$, $\mathrm{drive}(M)$, $\mathrm{drive}(D)$, $\mathrm{agent}(D, D)$, $\mathrm{agent}(D, M)$, $\mathrm{agent}(M, D)$, $\mathrm{agent}(M, M)$

H: $\exists x,y.\ \mathrm{guy}(x) \wedge \mathrm{agent}(y, x) \wedge \mathrm{drive}(y)$

Ground clauses: $\mathrm{guy}(M) \wedge \mathrm{agent}(D, M) \wedge \mathrm{drive}(D)$

Query Formula

[Beltagy and Mooney, StarAI 2014]

MLN's implementations calculates probabilities of ground atoms only

How to calculate probability of a complex query formula H ?

- Workaround

$$\mathrm{H} \leftrightarrow \mathrm{result}() \mid w = \boldsymbol{\infty}$$

$$\mathrm{P}(\mathrm{result}())$$

Query Formula
[Beltagy and Mooney, StarAI 2014]

Inference algorithm supports query formulas

$$P(H \mid KB) = \frac{Z(KB \cup \{(H, \infty)\})}{Z(KB)}$$

[Gogate and Domingos, 2011]

- Z: normalization constant of the probability distribution

Calculate Z: use SampleSearch [Gogate and Dechter, 2011]

- Works with mixed graphical models (probabilistic and deterministic)

Evaluation

[Beltagy and Mooney, StarAI 2014]

Dataset: SICK - RTE [SemEval, 2014]

	CPU Time (sec)	Timeouts (30 min)	Accuracy
MLN	147	96%	57%
M			
M			
MLN + Query + Speed	7	2.1%	72%

MLNs inference can be fast and efficient

Textual Entailment

[Beltagy et al., CompLing 2016]

Dataset: SICK - RTE [SemEval, 2014]

System	Accuracy
Logic	73.4%
Logic + precompiled rules + weight mapping + multiple parses	80.4%
Logic + Robinson resolution rules	83.0%
Logic + Robinson resolution rules + precompiled rules + weight mapping + multiple parses	85.1%
Current state of the art (Lai and Hockenmaier 2014)	84.6%

Textual Similarity

Semantic Textual Similarity (STS) [Agirre et al., 2012]

- Given two sentences $\mathrm{S}_1,\ \mathrm{S}_2$
- Evaluate their semantic similarity on a scale from 1 to 5

Example

- S_1: “A man is playing a guitar.”
- S_2: “A woman is playing the guitar.”
- score: 2.75

Example

- S_1: “A car is parking.”
- S_2: “A cat is playing.”
- score: 0.00

Textual Similarity — Settings

[Beltagy, Erk and Mooney, ACL 2014]

(proposal work)

Logical form

- CCG parser + Boxer

Knowledge base

- Precompiled rules: WordNet
- On-the-fly rules between all pairs of words

Inference

- $\mathrm{P}(\mathrm{S}_1|\mathrm{S}_2,\ \mathrm{KB}),\ \mathrm{P}(\mathrm{S}_2|\mathrm{S}_1,\ \mathrm{KB})$
- MLN and PSL inference algorithms suited for the task

PSL Relaxed Conjunction (for STS)

[Beltagy, Erk and Mooney, ACL 2014]

Conjunction in PSL (and MLN) does not fit STS

- T: A man is playing a guitar.
- H: A woman is playing the guitar.
- (score: 2.75)

Introduce a new "average operator" (instead of conjunction)

- $I(\ell 1 \wedge \ldots \wedge \ell n) = avg(\ I(\ell 1),\ \ldots,\ I(\ell n))$

Inference

- "average" is a linear function
- No changes in the optimization problem
- Heuristic grounding (details omitted)

Integrated into the official release of PSL

Evaluation – STS inference
[Beltagy, Erk and Mooney, ACL 2014]

Compare MLN with PSL on the STS task

	PSL time	MLN time	MLN timeouts (10 min)
msr-vid	8s	1m 31s	9%
msr-par	30s	11m 49s	97%
SICK	10s	4m 24s	36%

Apply MCW to MLN for a fairer comparison because PSL already has a **lazy grounding**

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
- Future work
- Conclusion

Question Answering

Open-domain Question Answering

- Given a document T and a query $\mathrm{H}(x)$
- Find the named entity e from T that best fills x in $\mathrm{H}(x)$

- T: …. The Arab League is expected to give its official blessing to the military operation on Saturday, which could clear the way for a ground invasion, CNN's Becky Anderson reported. The Arab League actions are …
- $\mathrm{H}(x)$: X blessing of military action may set the stage for a ground invasion

Inference: $\arg\max_x P(H(x)|T, KB)$

Question Answering

New challenges

- Long and diverse text
- Different inference objective

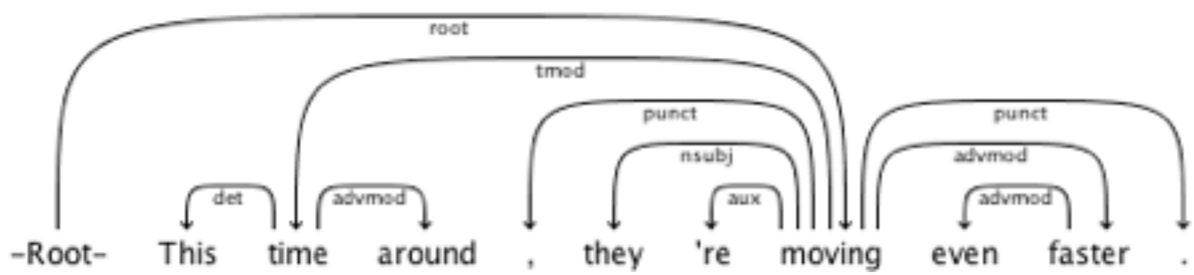
Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
  - Logical form
  - Knowledge base
  - Inference
- Future work
- Conclusion

Question Answering — Logical form

Translating dependency trees to Boxer-like output

- Rule-based translation
- More accurate
- Less expressive: no negation or quantifiers



$$\exists x,y,z,t.\ move(x) \wedge tmod(x,\ y) \wedge time(y) \wedge around(y) \wedge nsubj(x,\ z) \wedge they(z) \wedge adjmod(x,\ t) \wedge faster(t) \wedge even(t)$$

Question Answering — Logical form

Algorithm:

- Start from root, then iteratively for every relation do one of the following:
  - introduce new entity
  - merge with existing entity
  - ignore

Resulting logical form is a conjunction of predicates and relations

Limitation

- Does not represent any construct that requires “scope”
  - Negation
  - Quantifiers
  - Relative clauses

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
  - Logical form
  - Knowledge base
  - Inference
- Future work
- Conclusion

Question Answering — Knowledge base

On-the-fly rules — Robinson resolution rules

- assumes there is only one way to align T and H
- <u>not suitable for QA</u>

Question Answering — Knowledge base

On-the-fly rules — Graph-based alignment

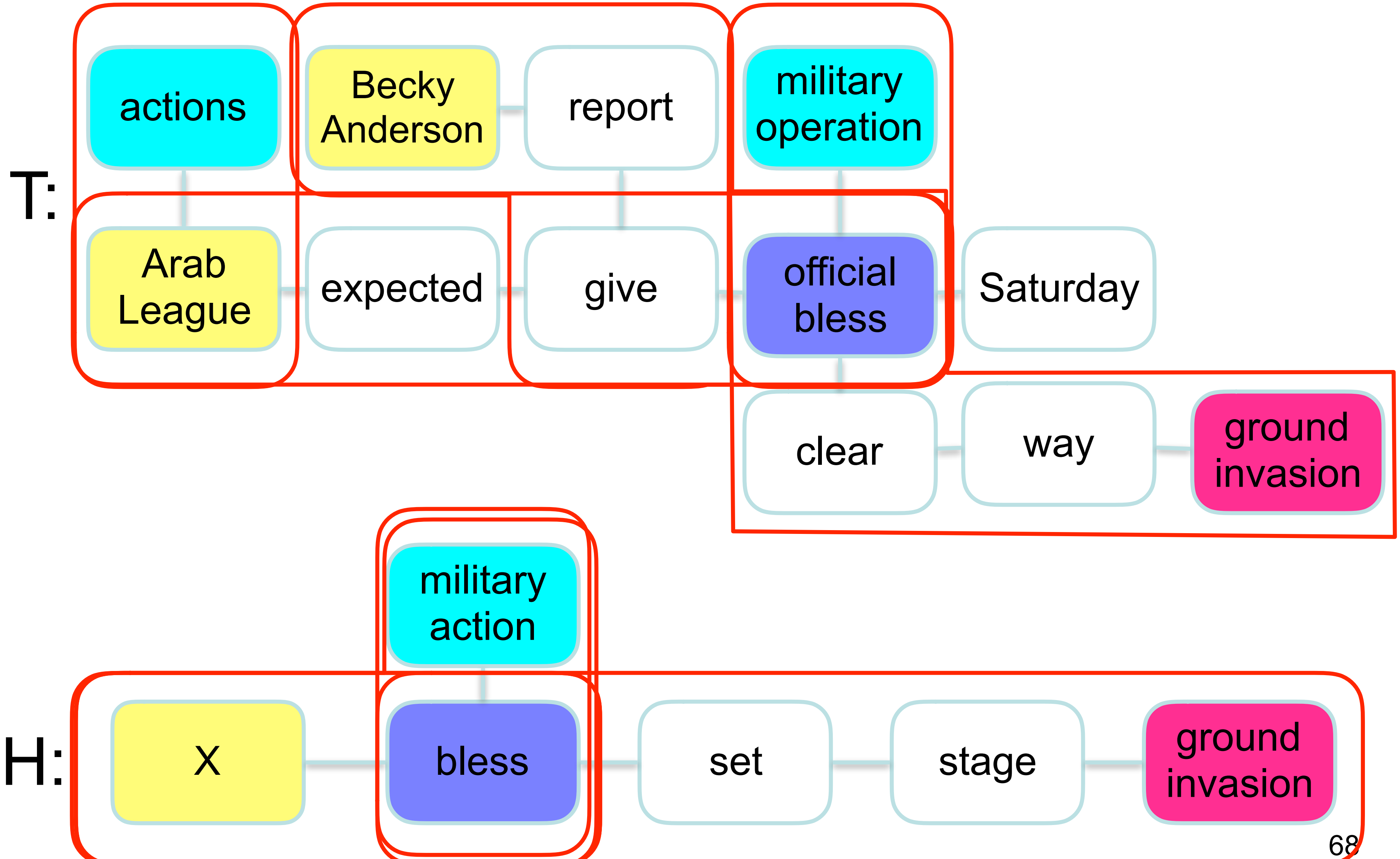
- view T and H as graphs
- align T and H based on a set of potentially matching entities
- extract rules from the alignment

Question Answering — Knowledge base

T: …. The Arab League is expected to give its official blessing to the military operation on Saturday, which could clear the way for a ground invasion, CNN's Becky Anderson reported. The Arab League actions are …

H: X blessing of military action may set the stage for a ground invasion

Question Answering — Knowledge base



Question Answering — Knowledge base

KB:

r1: Arab League expected to give official blessing ⇒ X blessing

r2: official blessing to military operation ⇒ blessing of military action

r3: official blessing clear way for ground invasion ⇒ blessing set stage for ground invasion

r4: Arab League actions ⇒ X blessing of military action

r5: Becky Anderson reported give official blessing ⇒ X blessing

Notes:

- Rules correspond to multiple possible alignments
- We have a procedure to automatically annotate the rules as positive and negative

Question Answering — Knowledge base

Annotating rules

- Run inference to find rules relevant to the right answer (positive rules). Remaining rules are negative rules
- Use the annotated rules to train a classifier to weight rules
- Repeat (Expectation Maximization)

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
  - Logical form
  - Knowledge base
  - Inference
- Future work
- Conclusion

Question Answering — Inference

Inference problem: $\arg\max_x P(H(x)|T, KB)$

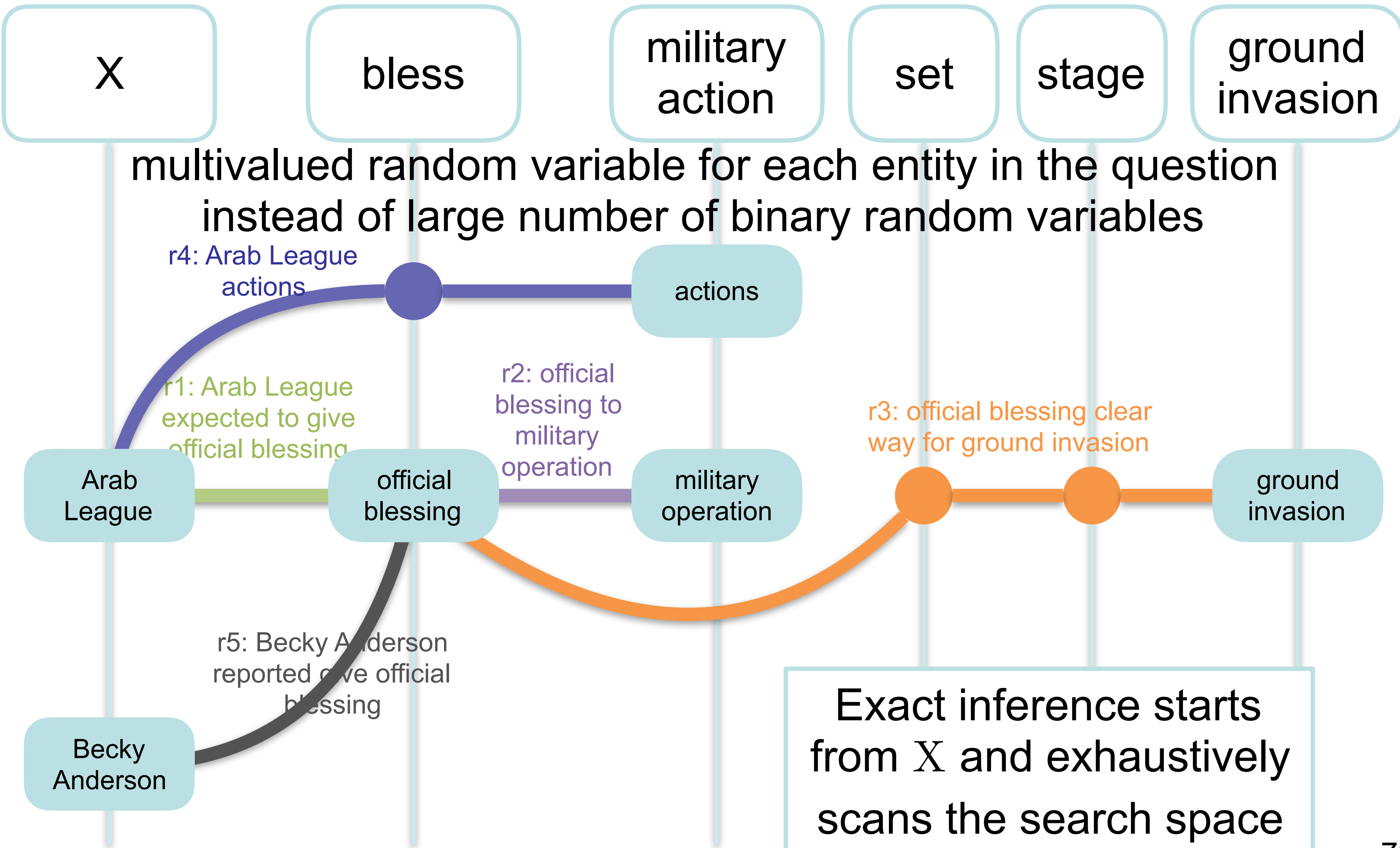
Can be solved using MLNs or PSL but they are not the most efficient

Define our own graphic model and its inference algorithm

- Encodes all possible ways of aligning the document and question
- Inference finds the best one

Question Answering — Inference

multivalued random variable for each entity in the question instead of large number of binary random variables



Exact inference starts from X and exhaustively scans the search space

Question Answering — Evaluation

Dataset:

- Collected from CNN (Hermann et al., 2015)
  - 380K training, 4K validation, 3K testing

System	Accuracy	Runtime
Preliminary PSL implementation	33%	4 seconds
This work	43%	9 milliseconds
This work + lexical entailment classifier	48%	
This work + alignment classifier	63%	
State of the art (Chen et al., 2016) — Neural Network	72%	

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
- Future work
- Conclusion

Future Work

Generalize QA implementation: inference as an alignment

- Logical form: learn the transformation of dependency tree to logical form to recover scope and other phenomena that dependency parsers do not support

- Generalize our graphic model formulation to other tasks

- Extend it to support negation and quantifiers

Future Work

Deep learning to integrate symbolic and continuous representations

Outline

- Introduction
- Logical form adaptations
- Knowledge base
- Question Answering
- Future work
- Conclusion

Conclusion

Probabilistic logic is a powerful representation that can effectively integrate symbolic and continuous aspects of meaning.

Our contributions include adaptations of the logical form, various ways of collecting lexical knowledge and several inference algorithms for three natural language understanding tasks.

☺

Multiple Parses

Reduce effect of mis-parses

Use the top CCG parse from

- C&C [Clark and Curran 2004]
- EasyCCG [Lewis and Steedman 2014]

Each sentence has two parses:

- Text: T1, T2
- Query: H1, H2

Run our system with all combinations and use the highest probability

Precompiled rules: WordNet

1) WordNet rules

- WordNet: lexical database of word and their semantic relations
- Synonyms: $\forall x.\ \mathrm{man}(x) \leftrightarrow \mathrm{guy}(x)$ | $w = \infty$
- Hyponym: $\forall x.\ \mathrm{car}(x) \rightarrow \mathrm{vehicle}(x)$ | $w = \infty$
- Antonyms: $\forall x.\ \mathrm{tall}(x) \leftrightarrow \neg\mathrm{short}(x)$ | $w = \infty$