Integrating Logical Representations with Probabilistic Information using Markov Logic

Dan Garrette, Katrin Erk, and Raymond Mooney

The University of Texas at Austin

Overview

- Some phenomena best modeled through **logic**, others **statistically**
- Aim: a unified framework for both
- We present first steps towards this goal
  - Basic framework: Markov Logic
  - Technical solutions for phenomena

Introduction

Semantics

- Represent the meaning of language
  - Logical Models
  - Probabilistic Models

Phenomena Modeled with Logic

- Standard first-order logic concepts
  - Negation
  - Quantification: universal, existential
- Implicativity / factivity

Implicativity / Factivity

- Presuppose truth or falsity of complement
- Influenced by polarity of *environment*

Implicativity / Factivity

- “Ed **knows** Mary left.”
    - ➡ Mary left
- “Ed **refused** to lock the door.”
    - ➡ Ed did not lock the door

Implicativity / Factivity

- “Ed did **not forget** to **ensure** that Dave failed.”
  - ➡ Dave failed
- “Ed **hopes** that Dave failed.”
  - ➡ ??

Phenomena Modeled Statistically

- Word Similarity
    - Synonyms
    - Hypernyms / hyponyms

Synonymy

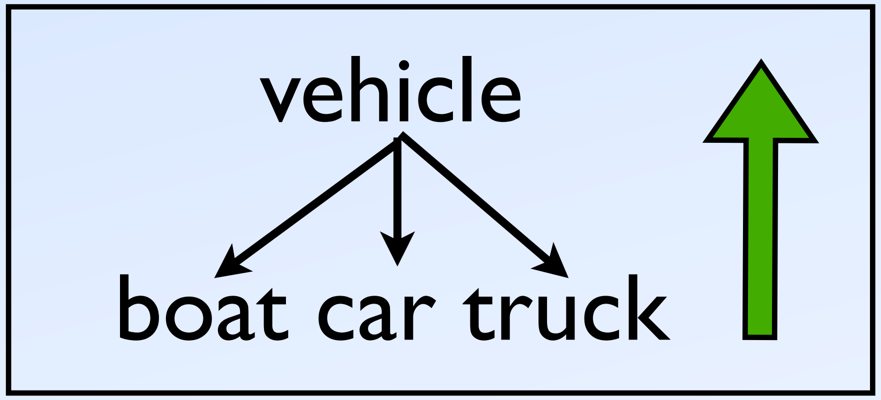
- “The wine **left** a stain.”
  - ➡ paraphrase: “result in”
- “He **left** the children with the nurse.”
  - ➡ paraphrase: “entrust”

Hypernymy

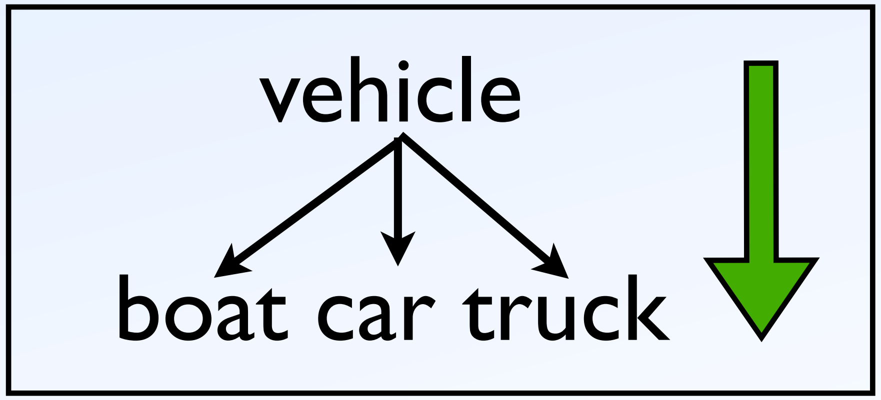
- “The **bat** flew out of the cave.”
  - ➡ hypernym: “animal”
- “The player picked up the **bat**.”
  - ➡ hypernym: “stick”

Hypernymy and Polarity

- “John owns a **car**”
  - ➡ John owns a **vehicle**



- “John does **not** own a **vehicle**”
  - ➡ John does **not** own a **car**



Our Goal

- A unified semantic representation
  - incorporate logic and probabilities
  - interaction between the two
- Ability to reason with this representation

Our Solution

- Markov Logic
- “Softened” first order logic: weighted formulas
- Judge *likelihood* of inference

Evaluating Understanding

- How can we tell if our semantic representation is correct?
- Need a way to measure comprehension
- *Textual Entailment*: determine whether one text implies another

Textual Entailment

premise:	iTunes software has seen strong sales in Europe.	Yes
hypothesis:	Strong sales for iTunes in Europe.	
premise:	Oracle had fought to keep the forms from being released	No
hypothesis:	Oracle released a confidential document	

Textual Entailment

- Requires deep understanding of text
- Allows us to construct test data that targets our specific phenomena

Motivation

Bos-style Logical RTE

- Generate rules linking ***all*** possible paraphrases
- Unable to distinguish between good and bad paraphrases

Bos-style Logical RTE

"The player picked up the **bat**."

$\models$ "The player picked up the **stick**"

$\not\models$ "The player picked up the **animal**"

Distributional-Only

- Able to judge similarity
- Unable to properly handle logical phenomena

Our Approach

- Handle logical phenomena discretely
- Handle probabilistic phenomena with weighted formulas
- Do both *simultaneously*, allowing them to influence each other

Background

Logical Semantics

- Semanticists have traditionally represented meaning with formal logic
- We use **Boxer** (Bos et al., 2004) to generate **Discourse Representation Structures** (Kamp and Reyle, 1993)

Logical Semantics

"John did not manage to leave"

x0

named(x0, john, per)

¬

e1 l2

manage(e1)
event(e1)
agent(e1, x0)
theme(e1, l2)
proposition(l2)

l2:

e3

leave(e3)
event(e3)
agent(e3, x0)

Logical Semantics

"John did not manage to leave"

Boxes have existentially quantified variables

...and atomic formulas

...and logical operators

x0

named(x0, john, per)

¬

e1 l2

manage(e1)
event(e1)
agent(e1, x0)
theme(e1, l2)
proposition(l2)

l2: e3

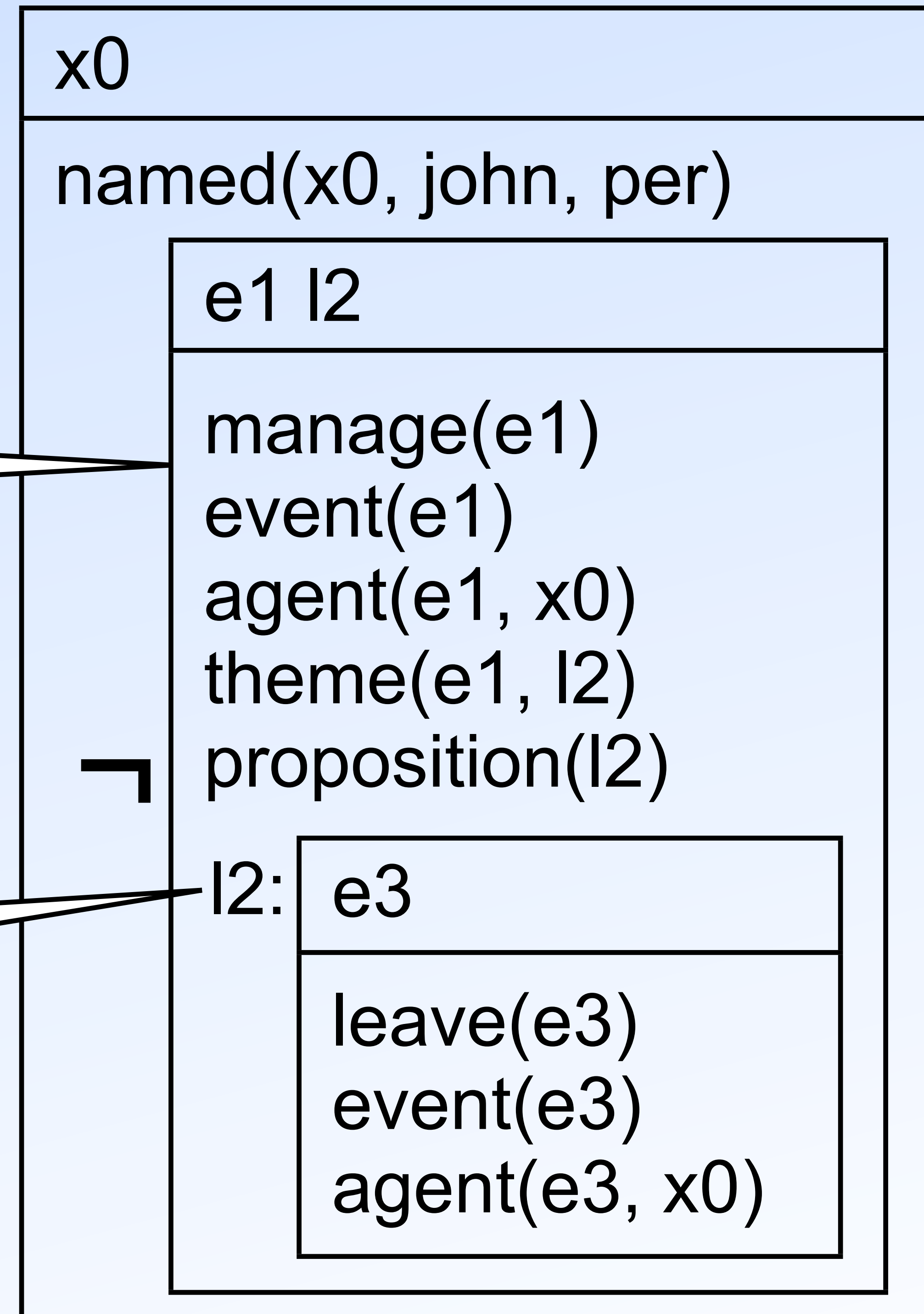
leave(e3)
event(e3)
agent(e3, x0)

Logical Semantics

"John did not manage to leave"

Box structure shows scope

Labels allow reference
to entire boxes



Logical Semantics

Why use First Order Logic?

- Powerful, flexible representation
- Straightforward inference procedure

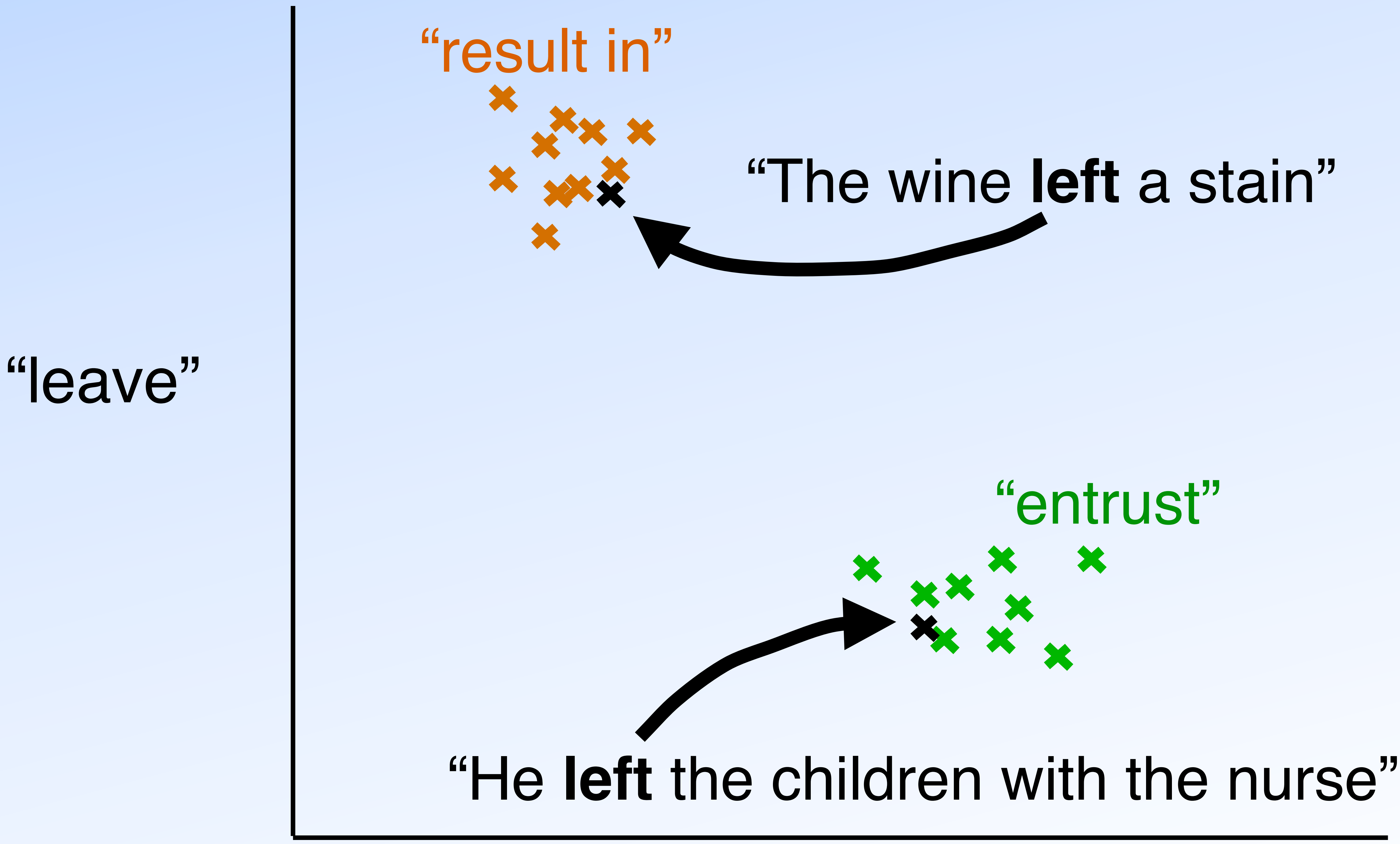
Why Not?

- Unable to handle uncertainty
- Natural language is not discrete

Distributional Semantics

- Describe word meaning by its context
- Representation is a continuous function

Distributional Semantics



Distributional Semantics

Why use Distributional Models?

- Can predict word-in-context similarity
- Can be learned in an unsupervised fashion

Why Not?

- Incomplete representation of semantics
- No concept of negation, quantification, etc

Approach

Approach

- Flatten DRS into first order representation
- Add weighted word-similarity constraints

Standard FOL Conversion

x0

named(x0, john, per)

¬ e1 l2

manage(e1)
event(e1)
agent(e1, x0)
theme(e1, l2)
proposition(l2)

l2: e3

leave(e3)
event(e3)
agent(e3, x0)

"John did not manage to leave"

$\exists$ x0.(ne_per_john(x0) &
¬ $\exists$ e1 l2.(manage(e1) &
event(e1) &
agent(e1, x0) &
theme(e1, l2) &
proposition(l2) &
$\exists$ e3.(leave(e3) &
event(e3) &
agent(e3, x0))))

Standard FOL Conversion

"John did not manage to leave"

```
x0
---------------------------
named(x0, john, per)

      e1 l2
      ----------------------
      manage(e1)
      event(e1)
      agent(e1, x0)
      theme(e1, l2)
 ¬    proposition(l2)

      l2: e3
          ------------------
          leave(e3)
          event(e3)
          agent(e3, x0)
```

DRT allows the theme proposition to be labeled as "l2"

∃ x0.(ne_per_john(x0) &
¬ ∃ e1 l2.(manage(e1) &
event(e1) &
agent(e1, x0) &
theme(e1, l2) &
proposition(l2) &
∃ e3.(leave(e3) &
event(e3) &
agent(e3, x0))))

The conversion loses track of what "l2" labels

Standard FOL Conversion

"John forgot to leave"

∃ x0 e1 l2.(ne_per_john(x0) &
forget(e1) &
event(e1) &
agent(e1, x0) &
theme(e1, l2) &
proposition(l2) &
∃ e3.(leave(e3) &
event(e3) &
agent(e3, x0)))

"John left"

∃ x0 e3.(ne_per_john(x0) &
leave(e3) &
event(e3) &
agent(e3, x0))

Standard FOL Conversion

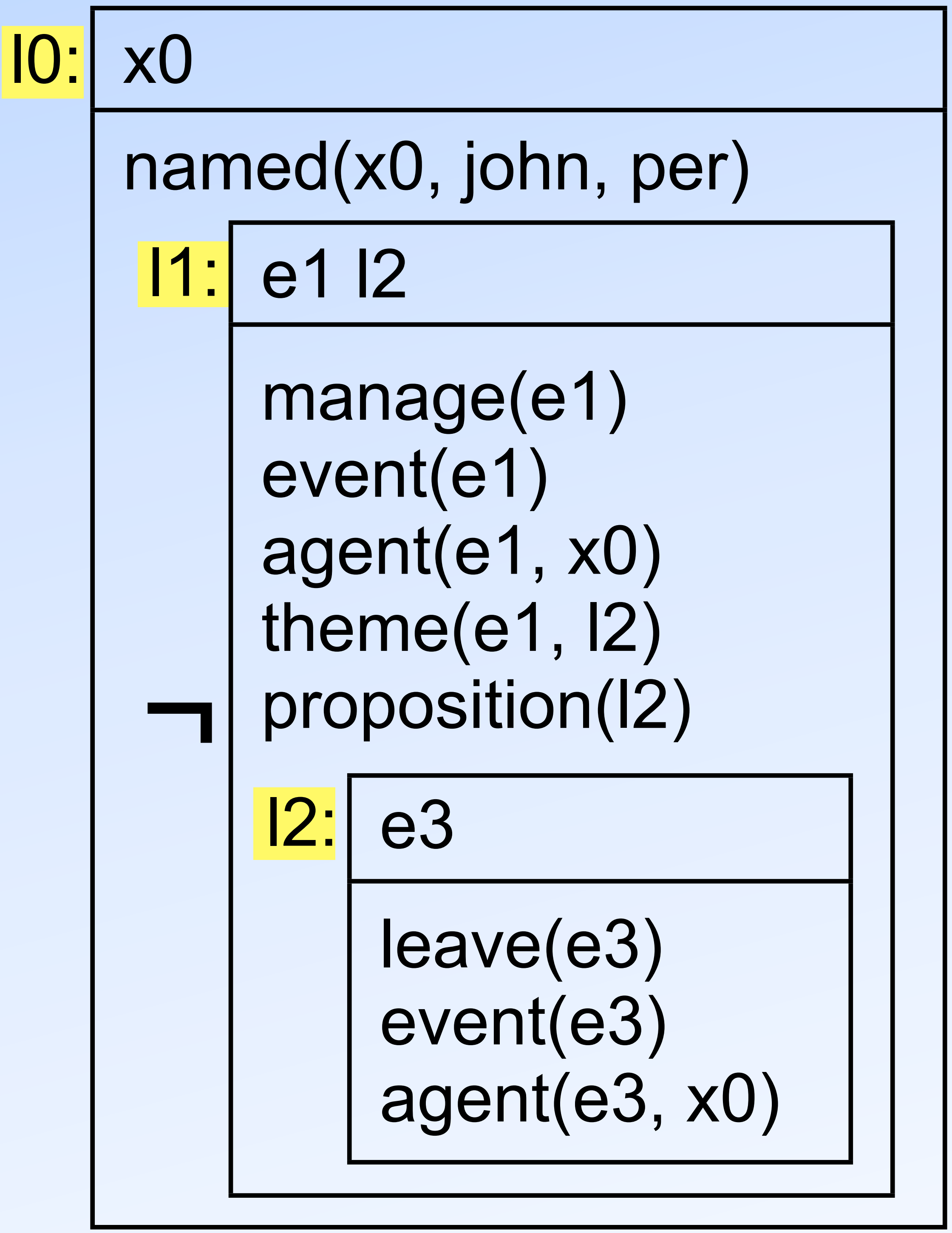
"John forgot to leave" $\not\models$ "John left"

$$\exists\, x0\ e1\ l2\ e3.(\text{ne\_per\_john}(x0)\ \&\ \text{forget}(e1)\ \&\ \text{event}(e1)\ \&\ \text{agent}(e1, x0)\ \&\ \text{theme}(e1, l2)\ \&\ \text{proposition}(l2)\ \&\ \text{leave}(e3)\ \&\ \text{event}(e3)\ \&\ \text{agent}(e3, x0))$$

$$\models$$

$$\exists\, x0\ e3.(\text{ne\_per\_john}(x0)\ \&\ \text{leave}(e3)\ \&\ \text{event}(e3)\ \&\ \text{agent}(e3, x0))$$

Our FOL Conversion



true(l0)

named(l0, ne_per_john, x0)

not(l0, l1)

pred(l1, manage, e1)
event(l1, e1)
rel(l1, agent, e1, x0)
rel(l1, theme, e1, l2)

prop(l1, l2)

pred(l2, leave, e3)
event(l2, e3)
rel(l2, agent, e3, x0)

label “l2” is maintained

Our FOL Conversion

- With "connectives" as predicates, rules are needed to capture relationships:

$$\forall\, p\ c.[(true(p) \wedge not(p,c)) \rightarrow false(c)]]$$
$$\forall\, p\ c.[(false(p) \wedge not(p,c)) \rightarrow true(c)]]$$

Implicativity / Factivity

- Calculate truth values of nested propositions
- For example, “forget to” is downward entailing in positive contexts:

$$\forall\, l1\ l2\ e.[(pred(l1, \text{“forget”}, e) \wedge true(l1) \wedge rel(l1, \text{“theme”}, e, l2)) \rightarrow false(l2)]$$

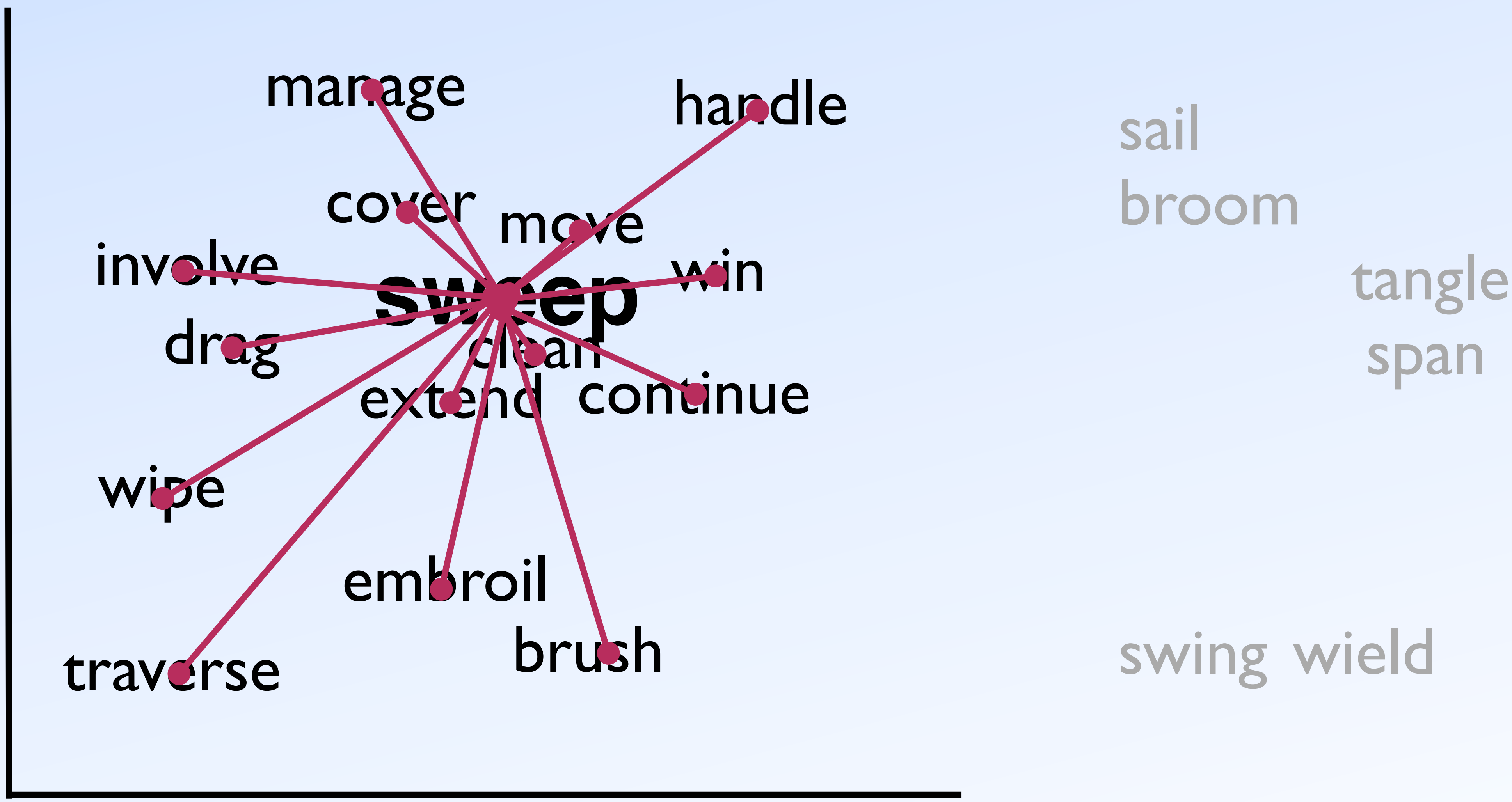
Word-Similarity

"A stadium craze is **sweeping** the country"

synset1: brush move
synset2: sail
synset3: broom wipe
synset4: embroil tangle drag involve
synset5: traverse span cover extend
synset6: clean
synset7: win
synset8: continue
synset9: swing wield handle manage

Word-Similarity

“A stadium craze is **sweeping** the country”



Word-Similarity

"A stadium craze is **sweeping** the country"

paraphrase	rank	P = 1/rank	W = log(P/(1-P))
continue	1	0.50	0.00
move	2	0.33	-1.00
win	3	0.25	-1.58
cover	4	0.20	-2.00
clean	5	0.17	-2.32
handle	6	0.14	-2.58
embroil	7	0.13	-2.81
wipe	8	0.11	-3.00
brush	9	0.10	-3.17
traverse	10	0.09	-3.32
sail, span, ...	11	0.08	-3.46

penalties increase with rank

Word-Similarity

"A stadium craze is **sweeping** the country"

- Inject a rule for every possible paraphrase
- MLN decides which to use

-2.00 $\forall$ l x.[pred(l, "sweep", x) $\leftrightarrow$ pred(l, "cover", x)]

-3.17 $\forall$ l x.[pred(l, "sweep", x) $\leftrightarrow$ pred(l, "brush", x)]

Evaluation

Evaluation

- Executed over 100 hand-written examples
- Hand-write examples instead of using RTE data to target specific phenomena
- Examples discussed in this talk are handled correctly by the system

Example

p: South Korea fails to honor U.S. patents

h_{good}: South Korea does not observe U.S. patents
h_{bad}*: South Korea does not reward U.S. patents

- “fail to” is negatively entailing in positive environments
- In context, “observe” is a better paraphrase than “reward”

Conclusion

Conclusion

- Presented unified logical/statistical framework for semantics
  - Markov Logic
  - Allows interaction between logic and probabilities
- Technical solutions for phenomena

Next Steps

- Large-scale evaluation
- Address a larger number of phenomena

Thank You!