Computational Linguistics

CSC 2501 / 485
Fall 2015

11

11. Anaphora; semantic representation; Wrap-up

Frank Rudzicz
Toronto Rehabilitation Institute-UHN, and
Department of Computer Science, University of Toronto

Reading: Jurafsky & Martin: 17.0–17.4.0, 17.5, 18.0–18.3.1.
Bird et al: *10.*

Anaphora and proforms

- ***Anaphora:*** Abbreviated backward reference in text.

  Anaphor: A word that makes an anaphoric reference.

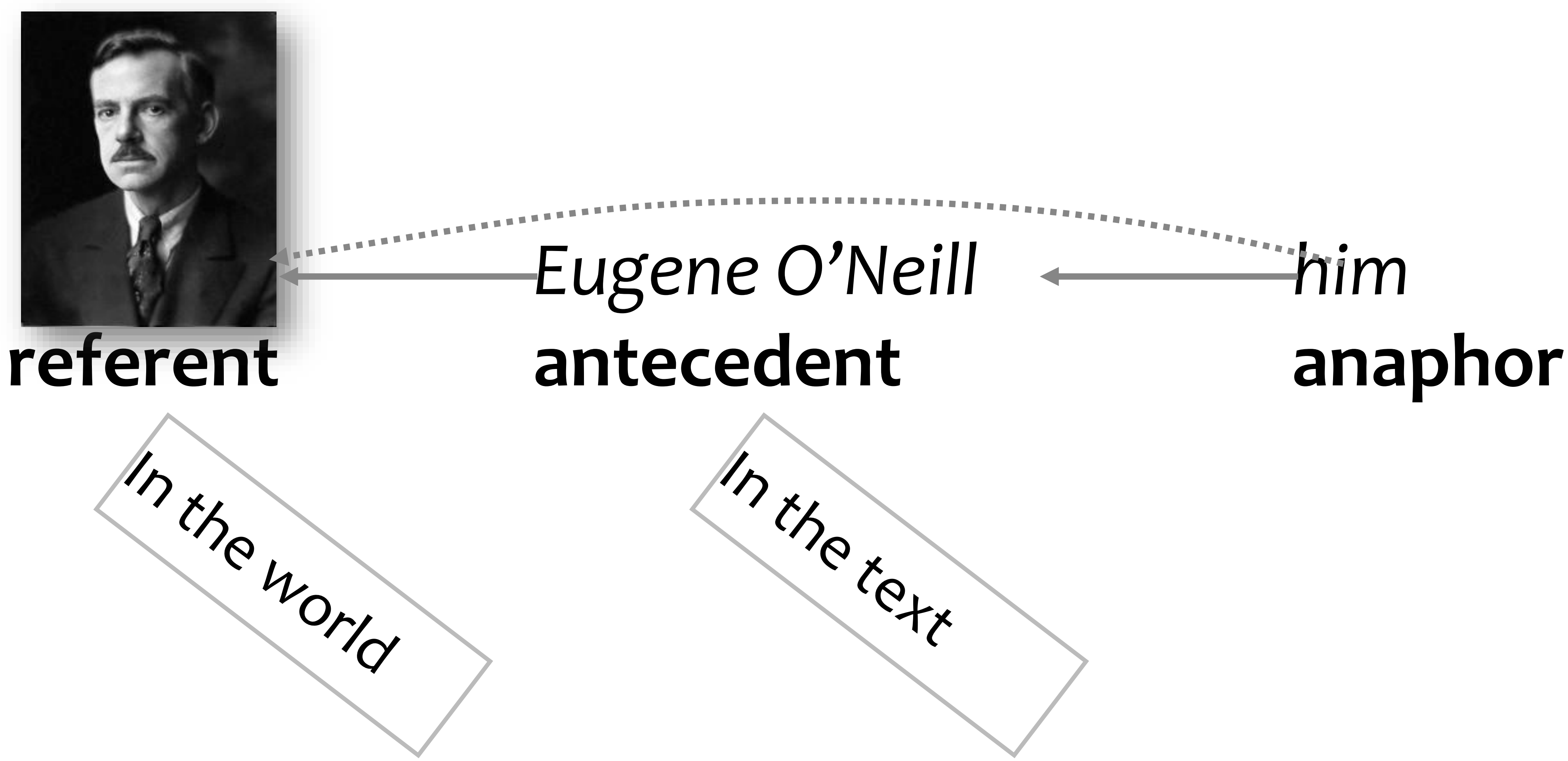
- ***Pronouns*** as canonical anaphors.

Eugene O'Neill was an American playwright. His plays involve characters who inhabit the fringes of society, engaging in depraved behavior, where they struggle to maintain their hopes and aspirations …

Example based on Wikipedia article on Eugene O'Neill, November 2007.

Reference and antecedence

- Loose usage: "referring back in text".
- Better terminology:



The anaphor and antecedent ***corefer.***

Photo from Wikipedia article on Eugene O'Neill, November 2007.

Constraining antecedents 1

- Pronouns must match antecedent in **number** and **gender**.
- Antecedent must be "nearby" in text.
  Hobbs 1978: 90% of antecedents are in same sentence as pronoun, 8% are in previous sentence.
- Antecedent must be "in focus".

?*Ross put the wine on the table$_i$. **It**$_i$ was brown and round.*

Constraining antecedents 2

- Syntactic contraints:

*Ross$_i$ rewarded {**him**$_j$ | **himself**$_i$} for **his**$_{i,j?}$ good work.*
*Nadia$_i$ said that **she**$_{i,j?}$ had done good work.*
***She**$_i$ said that **she**$_{i,j?}$ was pleased with Nadia$_k$.*

A (non-possessive) pronoun in an NP or PP that refers to the subject NP at the same level must be reflexive. A pronoun subject cannot corefer with a full NP at the same or lower level.

*Because **he**$_{i,j?}$ likes chocolate, Ross$_j$ bought a box of truffles.*

The antecedent of a forward-referring pronoun must **command** it: *i.e.*, the antecedent's immediately-dominating S node must **non-immediately** dominate the pronoun. *Exercise:* Draw this.

Anaphor resolution

- ***Anaphor resolution:*** Determining the antecedent or referent of an anaphor.
  - Antecedent might be another anaphor — a chain of coreferring expressions.
- Baseline algorithm: Choose most-recent NP that matches in gender and number.
  - Helpful tool: Gender guesser for names.

Some antecedent problems

- Composite or implicit antecedents:

*After Nadia$_i$ met Ross$_j$, **they**$_{\{i,j\}}$ went swimming.*
*Ross gave each girl$_i$ a crayon$_j$. **They**$_{\{i\}}$ used **them**$_{\{j\}}$ to draw pictures.*

- Antecedents that are events expressed as verbs or whole sentences:

*The St. Louis bank Page, Bacon & Co. **suspended operations** in February; **it** caused a panic in San Francisco.*

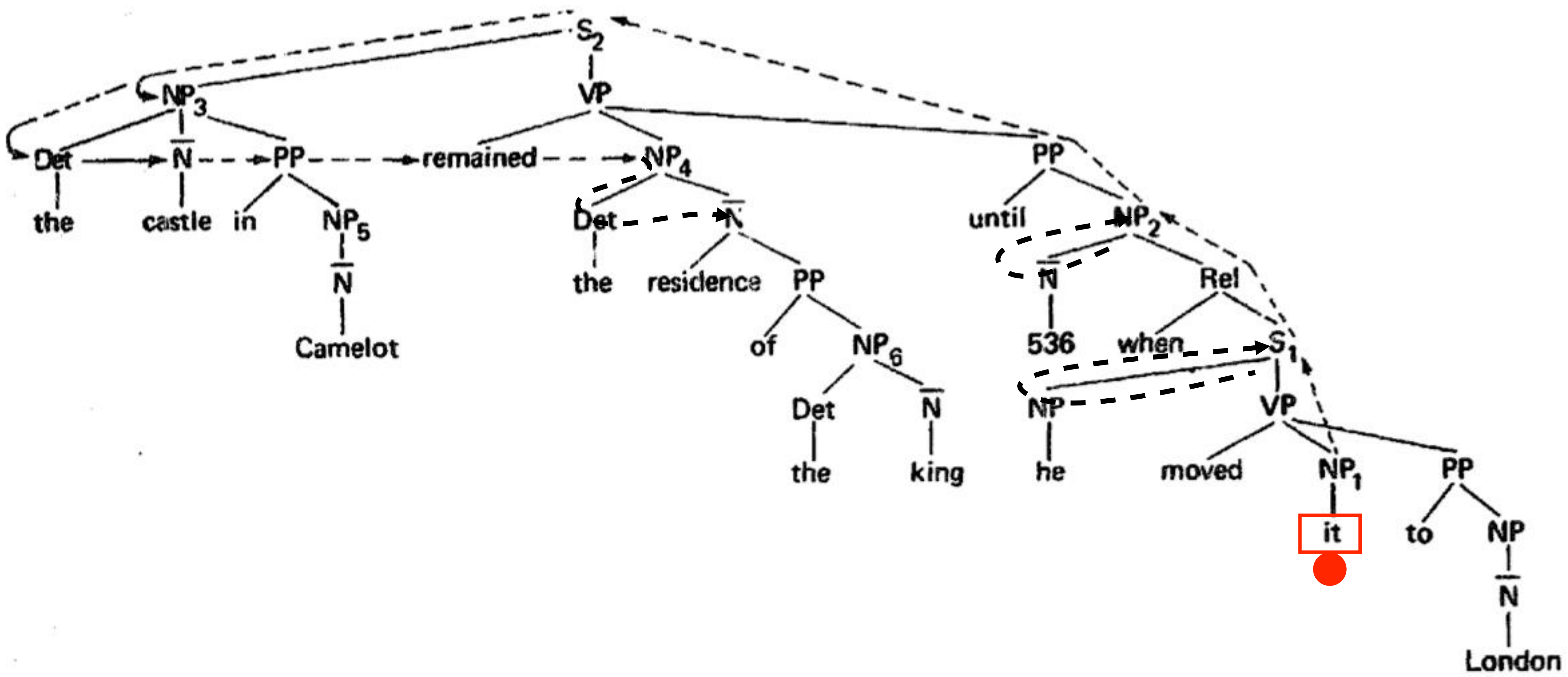
Based on "Banks, businesses cashed in by 'mining the miners.'" by Dale Kasler, 18 Jan 1998. http://www.calgoldrush.com/part4/04business.html

Hobbs's algorithm 1

- Traverse parse tree searching for candidate antecedents for pronouns.
- Choose first candidate NP that matches in gender and number
  (and maybe also in basic selectional preferences).
- Search order:
  - Start at pronoun, work upwards, scanning S and NP nodes left-to-right, breadth-first.
  - If necessary, traverse previous sentences, scanning S and NP nodes left-to-right, breadth-first.

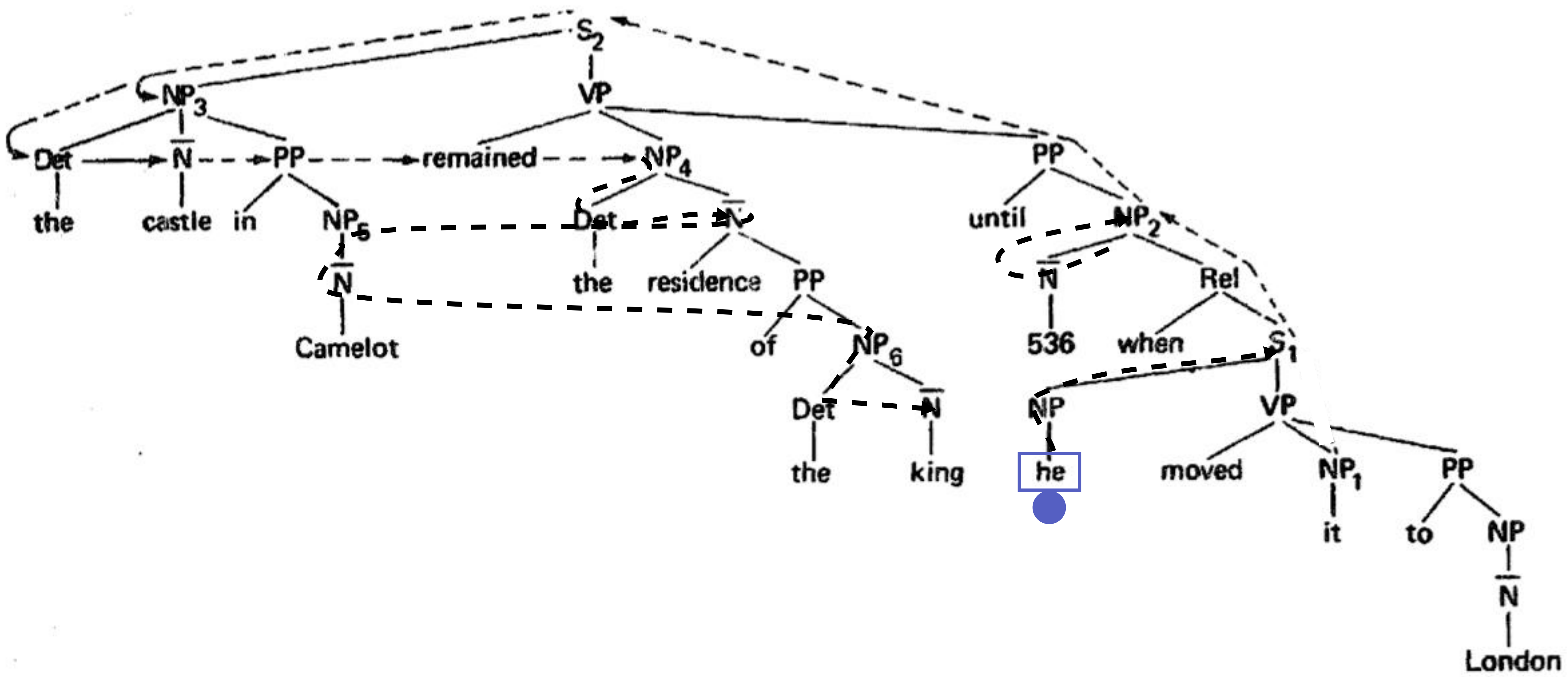
Hobbs, Jerry R. "Resolving pronoun references", *Lingua*, 44, 1978, 311–338.

Hobbs's algorithm 2



Adapted from: Hobbs 1978, figure 2.

Hobbs's algorithm 3



Adapted from: Hobbs 1978, figure 2.

Hobbs's algorithm 4

- Evaluation:
  - In 300 examples, it found correct antecedent 88% without selectional restrictions, 92% with.
  - But 168 had only one plausible antecedent.
    Performance on this subset was apparently 100%.
  - On the other 132, scores were 73%, 82% with and without selectional restrictions, respectively.

Soon, Ng, and Lim 2001 1

- A machine-learning approach.
- Goal: Find chains of coreferences in text (including definite references and anaphors).
- Basic idea:
  - Classify a pair of NPs in text as either coreferring or not.
  - Classifier is ***learned*** from data: text marked with positive and negative examples of coreference.
  - Features for classification are largely superficial, not syntactic.

Soon, Wee Meng; Ng, Hwee Tou; and Lim, Daniel Chung Yong. "A machine learning approach to coreference resolution of noun phrases." *Computational Linguistics*, 27(4), 2001, 521–544.

Soon, Ng, and Lim 2001 2

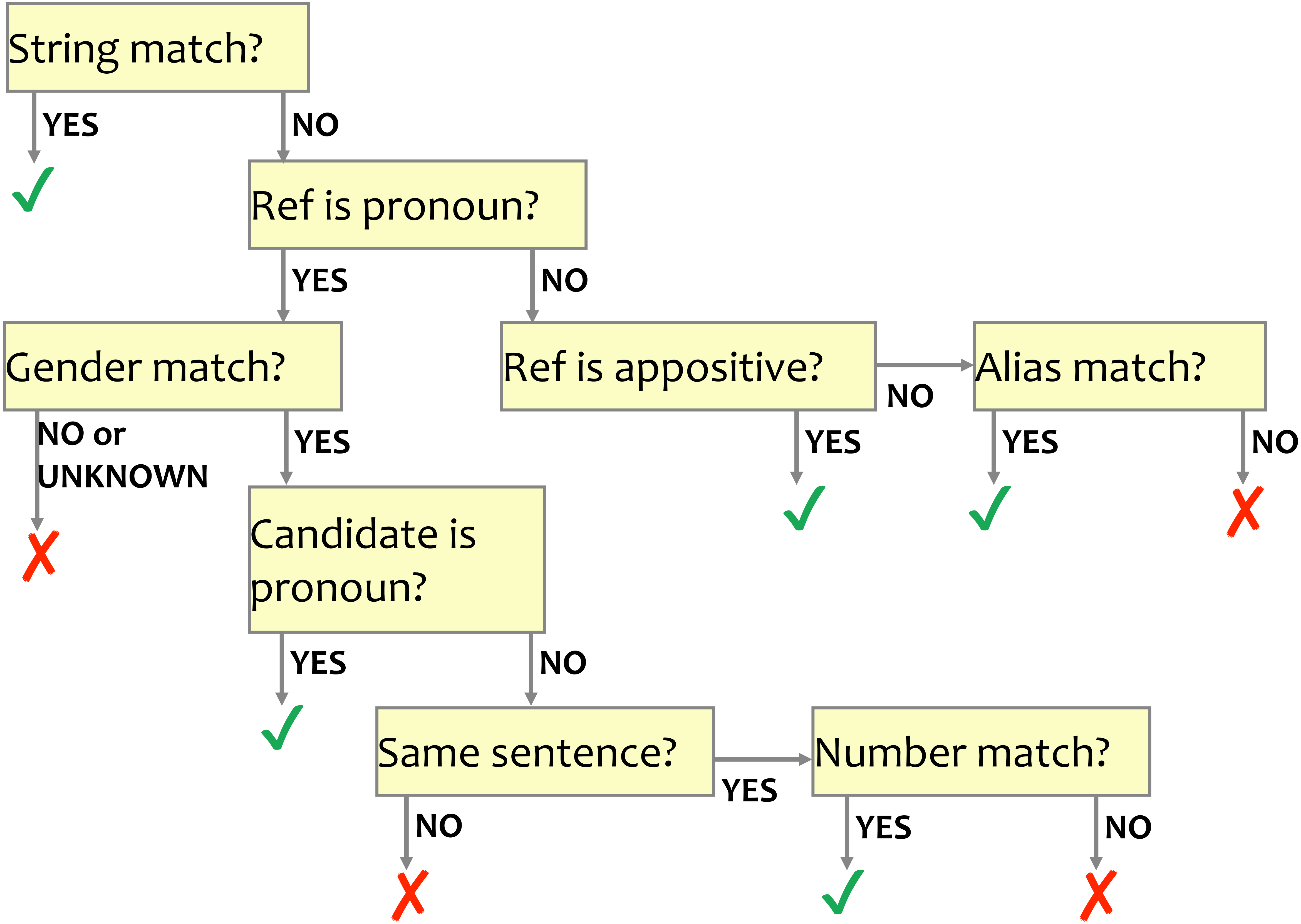
- Method:
  - Find “markables” in text (including nested ones, e.g. both '*CEO of Anaconda*' and '*Anaconda*').
  - For each one, work backwards through preceding markables until a coreferent is found
    (or give up by running out of candidates).
  - Yes/no decision on markable and candidate antecedents are made with decision-tree classifier induced from data by C5 algorithm.

Soon, Wee Meng; Ng, Hwee Tou; and Lim, Daniel Chung Yong. “A machine learning approach to coreference resolution of noun phrases.” *Computational Linguistics*, 27(4), 2001, 521–544.

Features for classification

- Distance apart (in sentences).
- Is either a pronoun?
- Is reference definite or demonstrative?
- String match or alias match?
  Bart Simpson, Mr Simpson; IBM, International Business Machines.
- Number, gender agreement?
- Semantic class agreement (by first sense in WordNet)?
  FEMALE, MALE, PERSON, ORGANIZATION. LOCATION, DATE, TIME, MONEY, PERCENT, OBJECT.
- Are both proper names?
- One is a proper name, and the reference is appositive?

Coreference classifier for MUC-6 data



Based on figure 2 of Soon et al., 2001.

Issues

- Accuracy depends on accuracy of pre-processing:
  - Finding markables (85%), determining semantic classes (???%).
- Not all features are used.
  - Semantic classes too inaccurate?
- String match and alias are important features, but are sometimes misleading.

Raghunathan et al’s sieve 1

- Avoid ML methods that consider all features at once.
  - Low-precision features may overwhelm high-precision features.
- Avoid methods that consider only one candidate at a time.
  - Might make selection too soon.
- Avoid supervised learning-based models.
  - Use knowledge-based heuristics.

Raghunathan, Karthik; Lee, Heeyoung; Rangarajan, Sudarshan; Chambers, Nate; Surdeanu, Mihai; Jurafsky, Dan; Manning, Christopher. “A Multi-Pass Sieve for Coreference Resolution.” *Proc, 2010 Conf on Empirical Methods in Natural Language Processing*, Cambridge, MA, 492–501.

Raghunathan et al's sieve 2

- "Sieve" is pipeline of rule-based modules.
- Considers all candidates simultaneously.
    - Starts with high-precision features, working through to low-precision.
- Mentions may be clustered (or added to cluster) even if not resolved.
    - Label clusters with gender, number, etc, as it becomes known.

Raghunathan et al's sieve 3

Modules (in order of decreasing precision):

- Exact match
- Appositives and similar, acronyms, demonyms.
- Strict head-matching (no non-matching stop words).
  - Matches *the Florida Supreme Court*, *the Florida court* but not *Yale University*, *Harvard University*.
- Pronoun matches (lexicon for gender, animacy).

Raghunathan et al's sieve 4

- Results: On MUC data, *PRF* = (.905, .680, .777).
  - High precision, moderate recall (as expected by design).
- Most noun recall errors are due to lack of semantic knowledge.
  - E.g., recognizing that *settlements* are *agreements*, *Gitano* is *the company*.

Adding world knowledge

- Some anaphors seem to need complex knowledge and inference to resolve.

The city councillors denied the demonstrators a permit because ***they*** *were communists.*

The city councillors denied the demonstrators a permit because ***they*** *feared violence.*

Winograd, Terry. *Understanding Natural Language*. Academic Press, 1972.

Winograd Schema 1

- Resolve anaphors with different antecedents in minimally-different sentence pairs.
- **World knowledge** and **inference** are required.

  The trophy would not fit in the brown suitcase because it was too big | small. What was too big?

- In practice: artificial problem, rarely seen in text; solution requires puzzle-solving as much as language understanding.

Levesque, Hector. The Winograd schema challenge. *Proc, AAAI- Spring Symposium on Logical Formalizations of Commonsense Reasoning*, 2011.

Winograd Schema 2

Nonetheless, Rahman and Ng:

- Special resolver for Winograd sentences.
- 73% accuracy on test set of 282 pairs.
  (Baseline 50%; conventional system ~55%.)
- From FrameNet, Google searches, and corpora: common narrative chains, connective relations, selectional restrictions.
- Cheap tricks or legitimate human-like method?

Rahman, Altaf and Ng, Vincent. Resolving complex cases of definite pronouns: The Winograd schema challenge. *Proc, 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju, 2012, 777–789.

It’s all about meaning

- **Semantic interpretation** or **conceptual analysis:**
  - Process of determining the *meaning* of a sentence or other utterance.
- What is semantic? What is meaning?
  - Many theories and views.
- What is meaning to a computer?

Req. of a semantic theory 1

- A semantic theory should explain what "meanings" are and how they operate.

  (An account of meaning that is adequate from one perspective may be quite unsuitable from another)

- A semantic theory should account for …
  - the meaning of *words*;
  - the meaning of *sentences*;
  - the *relationship* between the meaning of sentences and the meaning of the words in them.

Req. of a semantic theory 2

- A semantic theory should account for properties that “meanings” can have:
  - (non-syntactic) ambiguity
  - synonymy
  - vagueness
  - intensional reference
  - implication
  - …

Req. of a semantic theory 3

- A semantic theory should provide a ***representation*** for meaning that permits semantic interpretation:
  - Amenable to computations with respect to ambiguity, synonymy, etc.,
  - with the application of world knowledge as necessary.

Meaning as reference

- **“The meaning of an utterance is an object or event in the world.”**
- Problems:
  - Meaning ≠ reference.
    The morning star, the evening star.
  - Non-existent things.
    The first female president of the U.S.,
    a married bachelor
  - Syncategorematic words.
    and, if, therefore, …

Meaning as intent

- **“The meaning of an utterance is the intent of the speaker / writer.”**
- Large developed theory of ***speech acts***.

Meaning as behaviour

- **"The meaning of an utterance is the listener's or reader's behavioural response to it."**
- Procedure executed or action taken as a result of hearing the sentence uttered.
- Problems:
  - Action can be change of mental state; but how do we represent that?
  - Too dependent on individual?

Computational semantics

- What do *we* want to do with meanings?
  - Knowledge and information extraction from text.
  - Answering questions from text, knowledge base, or database.
  - Translation, interpretation, learning, acquisition of knowledge, …
  - *NL* → *meaning* and *meaning* → *NL.*

Semantic representations

- **Representation of knowledge** (KR) is a central problem of AI.
- Symbolic representations ("*vivid*"):
  - Logics, semantic nets and frames, executable procedures, …
- Distributed numeric weights ("*opaque*"):
  - Neural nets, vector-based methods, …

KR as semantics 1

- *Intuition:*
  - For people, meaning is something ‘in the world’ (as we represent it in our heads).
  - Words and sentences refer to objects, events, actions, ideas, etc, that we can perceive, apprehend, or carry out.
- A computer’s ‘world’ is a database or knowledge base, and the actions that it can execute.

KR as semantics 2

- We can represent **utterances** in the same formalism as **world knowledge**.
- Thus, the meaning of a sentence could be:
  - *Declarative:* A statement in a KR language that is to update or query a knowledge base.
  - *Procedural:* A segment of code to be executed to cause an effect, to update or query a database or knowledge base, etc.

Knowledge bases 1

- Two-part knowledge base:
  - **TBOX:** Definitions of terminology and necessary facts, including the basic ontology (hierarchy of object types).
  - **ABOX:** Contingent facts (possibly time-stamped as to when true).

Knowledge bases 2

- Three basic operations on KB:
  - **tell**: Assert a new fact to KB.
  - **retract**: Take statement out of KB (or mark as no longer true).
  - **ask**: Query whether statement is stored in KB, or for what value of variables the statement is true in KB.
- Argument is assertion in KR formalism.

KR as semantics 3

- Roles of KB:
  - *Repository of interpretation:*
    Interpreted input may be added to KB.
  - *Underlies representation of discourse structure:*
    Referents of recent mentions.
  - *Knowledge for interpretation:*
    Can be queried at any point in semantic analysis for facts, plausibility, etc.

Expressive power

- Most (all?) symbolic KR formalisms are ***first-order*** or less:
  - Equivalent to *first-order predicate calculus* (FOPC)*, with quantifiers ∀ and ∃.
  - Much less expressive than natural language.
- Need (at least) higher-order intensional alethic deontic epistemic temporal (modal) logics for full NL expressiveness.
- However, …

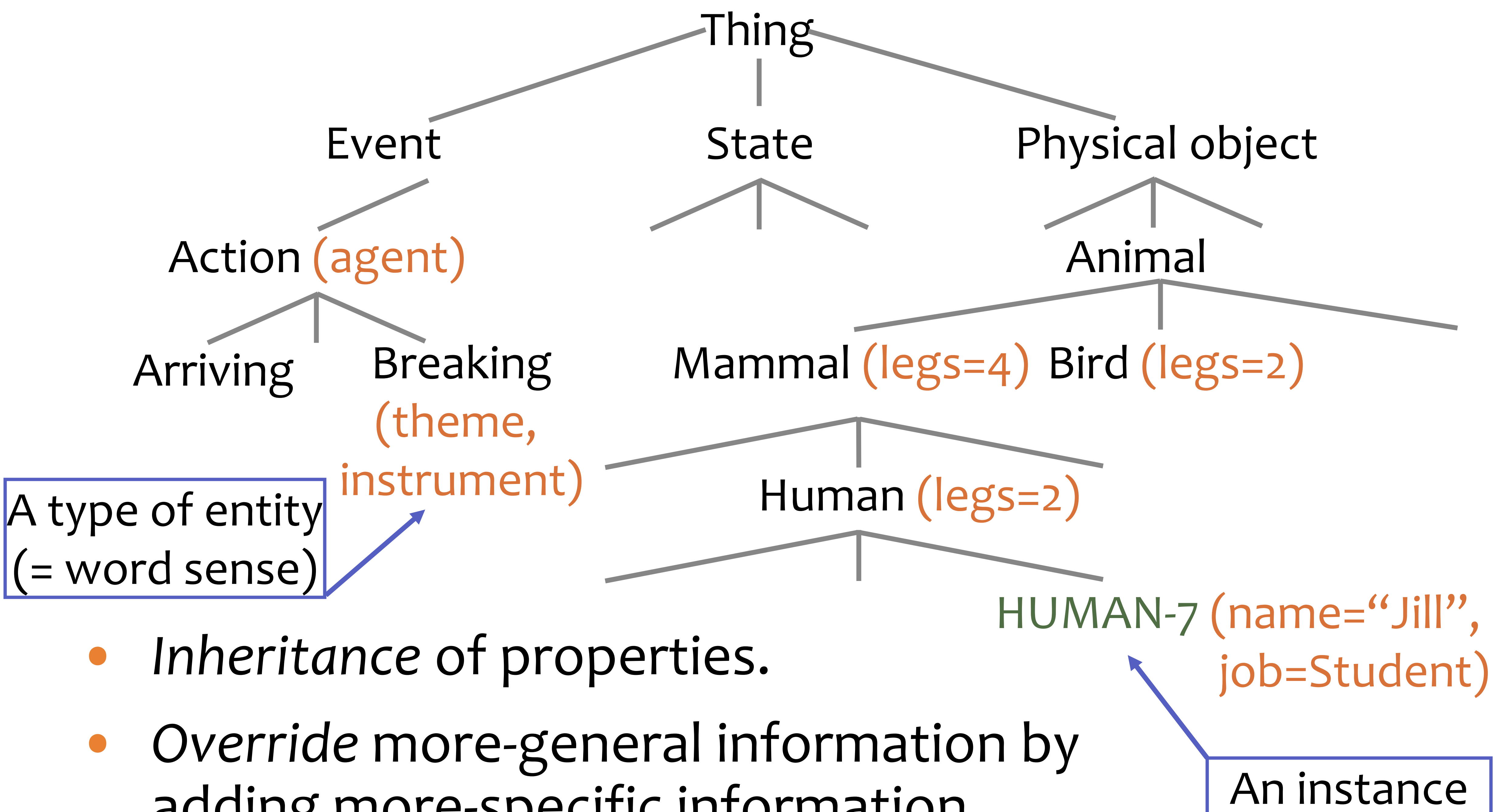
*A.k.a. *first-order logic* (FOL).

Frames and networks

- A typical approach:
  - Hierarchical classification of events and entities: taxonomy, ontology, is-a hierarchy.
  - Each node describes either:
    - a ***type*** of event or entity — in effect a potential ***word sense*** (≈ TBOX).
    - an ***instance*** of a type (≈ ABOX).
  - Descriptions in terms of attribute-value pairs.

A node need not have a corresponding word in any language.

Example



- *Inheritance* of properties.
- *Override* more-general information by adding more-specific information.

Example

```
HUMAN-7:
   instance-of: Human
   name: “Jill”
   legs: 2
   mother: HUMAN-23
   father: HUMAN-111
   job: Student
   address: (unknown)
```

OR

```
∃ HUMAN-7 (Human(HUMAN-7) ∧ name(HUMAN-7, “Jill”)
   ∧ legs(HUMAN-7, 2) ∧ mother(HUMAN-7, HUMAN-23)
   ∧ father(HUMAN-7, HUMAN-111)
   ∧ job(HUMAN-7, Student)
   ∧ address(HUMAN-7, (unknown)) )
```

First-order representation 1

- Constants represent instances denoted by names or definite references:

  `nadia, human-7, dog-16, wsptwe`

- One-place predicates represent properties denoted by nouns and adjectives.

  `dog(dog-16), happy(dog-16)`

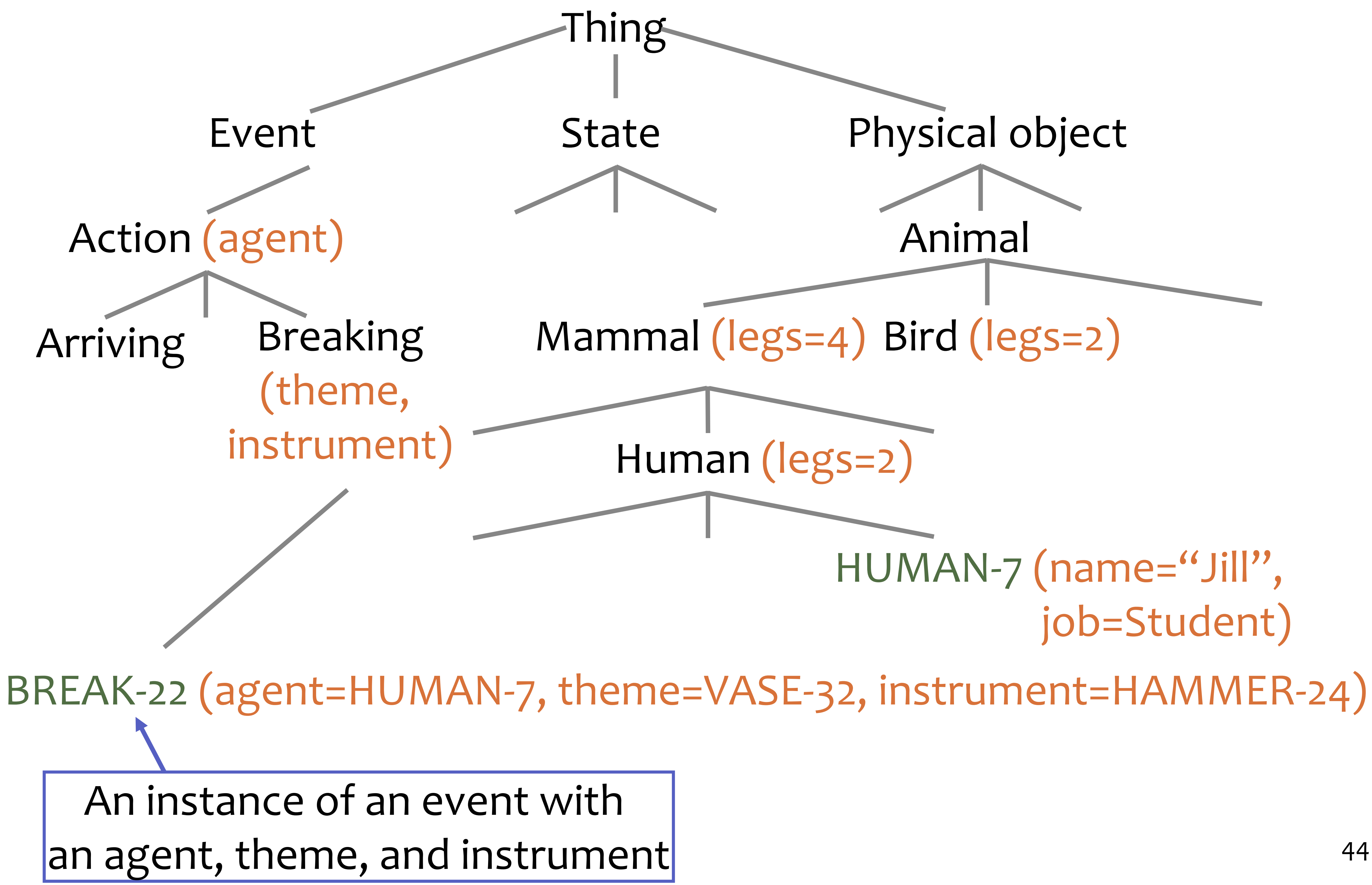
- Two-place predicates represent relational attributes:

  `owner(dog-16, human-7)`

The structure of events

- As important as the structure of physical objects.
- Attributes are **thematic roles**:
  - ***Agent:*** doer of the action.
  - ***Patient/Theme:*** entity affected by the action.
  - ***Instrument:*** entity used to do the action.
  - ***Result:*** entity created.
- **Adjunct or modifier** attributes include:
  - ***Time, Location, Manner.***

Example


Thing
Event
State
Physical object
Action (agent)
Animal
Arriving
Breaking (theme, instrument)
Mammal (legs=4)
Bird (legs=2)
Human (legs=2)
HUMAN-7 (name="Jill", job=Student)
BREAK-22 (agent=HUMAN-7, theme=VASE-32, instrument=HAMMER-24)
An instance of an event with an agent, theme, and instrument

Example

```
BREAK-22:
    instance-of: Breaking
    agent: HUMAN-7
    theme: VASE-32
    instrument: HAMMER-24
```

"Jill broke the vase with the hammer."

OR

```
∃ BREAK-22 (Breaking(BREAK-22)
    ∧ agent(BREAK-22, HUMAN-7)
    ∧ theme(BREAK-22, VASE-32)
    ∧ instrument(BREAK-22, HAMMER-24) )
```

OR

```
Breaking(BREAK-22, HUMAN-7, VASE-32, HAMMER-24)
```

Positional specification of arguments

First-order representation 2

- *n*-place or (*n*+1)-place predicates to represent relationships denoted by verbs.
  - ```
    sleep(nadia)
    cuddle(nadia, dog-16)
    give(nadia, dog-16, ross)
    ```
  - sleep(**sleep-23**, nadia)
cuddle(**cuddle-12**,nadia, dog-16)
give(**give-333**, nadia, dog-16, ross)
- We can decompose things into conjunctions of one- or two-place predicates.
    ```

First-order representation 3

- Interpreting NL sentences in first-order representations.

Nadia feeds Ross $\longrightarrow$

```
feed(nadia,ross)
```

OR

```
∃e (feeding(e) ∧ agent(e,nadia)
        ∧ theme(e,ross))
```

OR (Jurafsky & Martin)

```
∃e (feeding(e) ∧ feeder(e,nadia)
        ∧ fed(e,ross))
```

First-order representation 4

- Sentences with quantifiers:

{ All cows | Cows } eat ice-cream ⟶

```
∀x(cow(x) ⇒ eats(x,ice-cream))
```

Every student feeds a weasel ⟶

```
∀x(student(x) ⇒
       ∃y(weasel(y) ∧ (feeds(x,y))
```

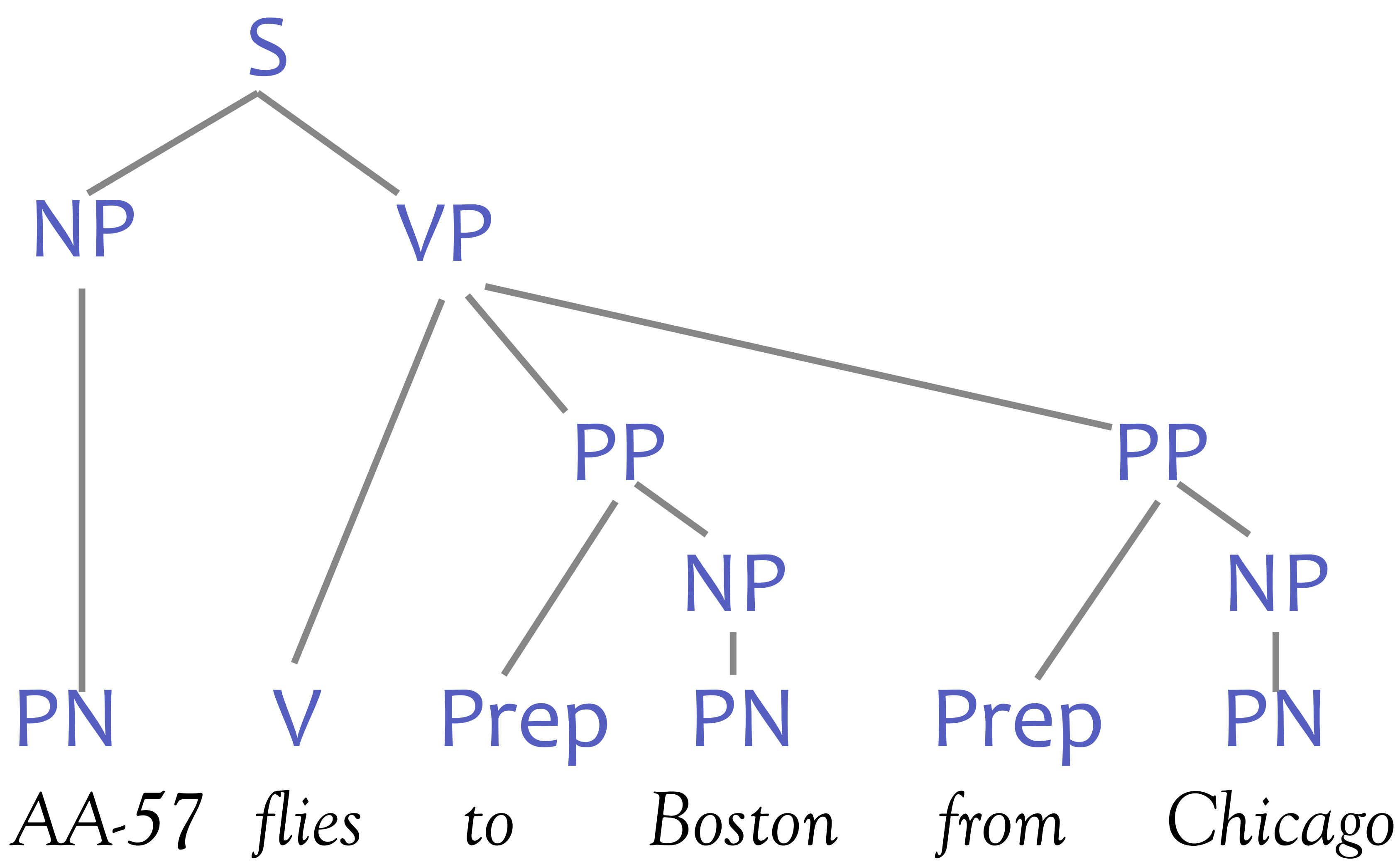
Ambiguity!

```
∃y(weasel(y) ∧
     ∀x(student(x) ⇒ feeds(x,y))
```

Procedural semantics

- **Input**: Parse tree.
- **Output:** Executable code to query database.
  - If process fails, try a different parse.
- Basic method: Pattern-matching rules:
  - If $P \subset tree$, insert $f(P)$ in output.

Example




```
connect(aa-57, chicago, boston)
```

Quantification

- System has four levels of rules:
  determiners, NPs, PPs on NPs, clauses.
- Rules for order of interpretation and for where result is placed cause NP quantifiers to apply to whole sentence.

  Every → (∀x: R; P)

- *Every flight from Boston to NY leaves Boston at 8:00.*

```
(∀x1 / flight: connect(x1, Boston, NY);
     equal (dtime (x1, Boston), 800))
```

Pros and cons

- Simple method.
  - Fast, but very superficial.
- May be good in very limited domains.
- Need to characterize anticipated input fairly specifically; not portable.
- Rules may interact in complex ways.
- Can’t detect ambiguity.

Methods of interpretation 2

- ***Semantic grammars:***
  - Tailor grammar to specific domain.
  - Integrate parsing and semantic analysis:
    parser builds semantic structure directly.
  - Grammar rules take semantic class into account.

    NP → City | Flight | Airline | …
    Flight → Flight-code | Time *flight to* City | …
    Time → *morning* | … | Number {*a.m.* | *p.m.*} | …
  - More-specific rules (less portable).

Methods of interpretation 3

- ***Compositional semantics***
- ***Principle of compositionality:***
  - The meaning of a syntactic constituent is a ***systematic function*** of the meaning of its parts.
- Philosophical and technical problems with compositionality:
  - What counts as “systematic”?
  - What about sense modulation (*fast typist*, *fast road*)?
  - What about the role of the *context* of a utterance?

Compositionality 1

- ***Rule-to-rule principle:***
  - Each syntactic rule has corresponding semantic rule.
  - Work in parallel or lockstep to build parse tree and ***logical form*** simultaneously.
- **Implication for semantic representation:** Representations must be combinable in various ways.
  - Representations of *Nadia* and *see* must be combinable (in different ways) to give representations of both *see Nadia* and *Nadia sees*.

Compositionality 2

- **Implication for grammar:**
  - Whenever there is a syntactic rule that combines two or more constituents to create a new one, …
  - … there is a corresponding semantic rule that creates the semantic interpretation for the new constituent from the interpretations of its components …
  - … by combination of one with the other.
  - *E.g.*, rule VP $\rightarrow$V NP must specify how the semantics of the V and NP combine to give semantics of the VP.

Semantic objects

- *Words:* Each word's entry in the lexicon has an associated ***semantic object*** from the KB or semantic representation.
- *Notation:* Use 'prime' on word or constituent to denote its associated semantic object:
VP′, *Nadia*′, [*Nadia sleeps*]′.

Combining semantic objects

- Regard one semantic object as a predicate, the other as its argument.
- *Example:* S → NP VP {Sem: S′ = VP′(NP′)}
  - VP″s must be functions of one argument.
  - [*Nadia sleeps*]′ = *sleeps*′(*Nadia*′) = *sleeps*′(`nadia`)
  - = `∃x(sleeping(x) ∧ sleeper(x,nadia))`
  - Hence *sleeps*′ is a function of one argument, a **lambda abstraction**:

    `λy(∃x(sleeping(x) ∧ sleeper(x,y)))`

Determiners & quantifiers 1

- [*Nadia feeds a fish*]′

$= \exists f (\text{fish}(f) \land \exists x (\text{feeding}(x) \land \text{feeder}(x,\text{nadia}) \land \text{feedee}(x,f)))$

a fish behaves quite differently than *Ross*

- NP → Det N {Sem: NP′ = Det′(N′)}

$a' = \lambda P \lambda Q (\exists f (P(f) \land Q(f)))$

$\textit{fish}' = \lambda a (\text{fish}(a))$

$[\textit{a fish}]' = \lambda Q (\exists f (\text{fish}(f) \land Q(f)))$

Determiners & quantifiers 2

feeds′ = $\lambda R \lambda z\ R(\lambda y(\exists x(feeding(x)$
$\wedge\ feeder(x,z)\ \wedge\ feedee(x,y))))$

[*a fish*]′ = $\lambda Q(\exists f(fish(f)\ \wedge\ Q(f)))$

[*feeds a fish*]′ = ***feeds***′([*a fish*]′)

$= \lambda z(\exists f(fish(f) \wedge \exists x(feeding(x)$
$\wedge\ feeder(x,z) \wedge feedee(x,f)))$

Determiners & quantifiers 3

- [*A fish feeds Ross*]′

```
= ∃f(fish(f) ∧ ∃x(feeding(x)
    ∧ feeder(x,f) ∧ feedee(x,ross)))
```

a fish behaves same as when object of sentence

Quantifier scope ambiguity 1

> *No man is an island, entire of itself;*
> *every man is a piece of the continent, a part of the main.*
> –John Donne, Meditation XVII, 1623

Every man is a piece of the continent ⟶

✓ `∀x(man(x) ⇒ ∃y(piece-of-continent(y) ∧ be(x,y)))`

✗ `∃y(piece-of-continent(y) ∧ ∀x(man(x) ⇒ be(x,y)))`

Quantifier scope ambiguity 2

> *He was inside the building when the rear of the plant exploded. "Everybody saw a huge fireball, and everybody started running out," Bales said.*
>
> –Associated Press, 21 Feb 2003

Every person saw a fireball ⟶

✗ $\forall x(\text{person}(x) \Rightarrow \exists y(\text{fireball}(y) \wedge (\text{saw}(x,y)))$

✓ $\exists y(\text{fireball}(y) \wedge \forall x(\text{person}(x) \Rightarrow \text{saw}(x,y)))$

Quantifier scope ambiguity 3

- Syntax doesn't help choose: parse tree has the same structure in each case.
- Present rules will give wide-scope reading.
  `∀x(...(x) ⇒ ∃y(...(y) ∧ ...(x,y)))`
- Need to make both choices available for separate disambiguation process.

Evaluating interpretations

- Do the semantic representations allow other processes to "do the right thing"?
  - Inference, retrieval, question-answering, …
- "Learning by reading"
  - Read 133 sentences from a high-school chemistry text and answer the exercises.

Rutu Mulkar, Jerry R. Hobbs, Eduard Hovy, Hans Chalupsky, and Chin-Yew Lin. Learning by reading: Two experiments. *Proceedings of the 3rd International Workshop on Knowledge and Reasoning for Question Answering,* Hyderabad, India, January 2007.

Some knowledge from book:

$H3O+$ is the conjugate acid of $H2O$.

Acids cause certain dyes to change color.

Bases have a bitter taste and feel slippery.

Soap is a base.

Exercises:

Does $H3O+$ cause certain dyes to change color?

Answer: yes

What taste does soap have?

Answer: bitter

Soap feels how?

Answer: slippery

Wrap up

What was in this course

- Intro to computational linguistics.
- Grammars and parsing; features.
- Chart parsing.
- Statistical PP attachment.
- Part-of-speech tagging; statistical parsing.
- Semantics and semantic interpretation.
- Lexical semantics.
- Word sense disambiguation.
- Neural word representations.
- Anaphora and coreference.

Syntax

Semantics

Underlying themes

- Ambiguity is a pervasive problem.
- Algorithms for parsing, interpretation, ambiguity resolution.
- Supplementing linguistic knowledge with statistical knowledge from corpora.
- Importance of lexical information.
- Reliance on semantic representations.

What's in CSC 401 / 2511

- Introduction to corpus-based linguistics.
- Text categorization, classification methods.
- N-gram models and smoothing.
- Entropy and information theory.
- (Hidden) Markov models.
- Statistical machine translation.
- Automatic speech recognition and synthesis.
- Information retrieval.
- Text summarization.

Thanks