Introduction to NLP

363.

Knowledge Representation

Knowledge Representation

- Ontologies
- Categories and objects
- Events
- Times
- Beliefs

Knowledge Representation

- Object
  - Martin the cat
- Categories
  - Cat
- Ontology
  - Mammal includes Cat, Dog, Whale
  - Cat includes PersianCat, ManxCat
- ISA relation
  - ISA (Martin,Cat)
- AKO relation
  - AKO (PersianCat,Cat)
- HASA relation
  - HASA (Cat, Tail)

Semantics of FOL

- FOL sentences can be assigned a value of *true* or *false.*

  ISA(Milo,Cat) = true

- *Milo is younger than Martin*

  <(AgeOf(Milo),AgeOf(Martin)) = true

  =(AgeOf(Milo),AgeOf(Martin)) = false

Examples with Quantifiers

- All cats eat fish

  $\forall x{:}ISA(x,Cat) \Rightarrow EatFish(x)$

Representing Events

- Martin ate
- Martin ate in the morning
- Martin ate fish
- Martin ate fish in the morning

One Possible Representation

- FOL representations
  - Eating1(Martin)
  - Eating2(Martin,Morning)
  - Eating3(Martin,Fish)
  - Eating4(Martin,Fish,Morning)
- Meaning postulates
  - Eating4(x,y,z) -> Eating3(x,y)
  - Eating4(x,y,z) -> Eating2(x,z)
  - Eating4(x,y,z) -> Eating1(x)

Example from Jurafsky and Martin

Second Possible Representation

- Eating4(x,y,z)
  - With some arguments unspecified
- Problems
  - Too many commitments
  - Hard to combine Eating4(Martin,Fish,z) with Eating4(Martin,y,Morning)

Example from Jurafsky and Martin

Third Possible Representation

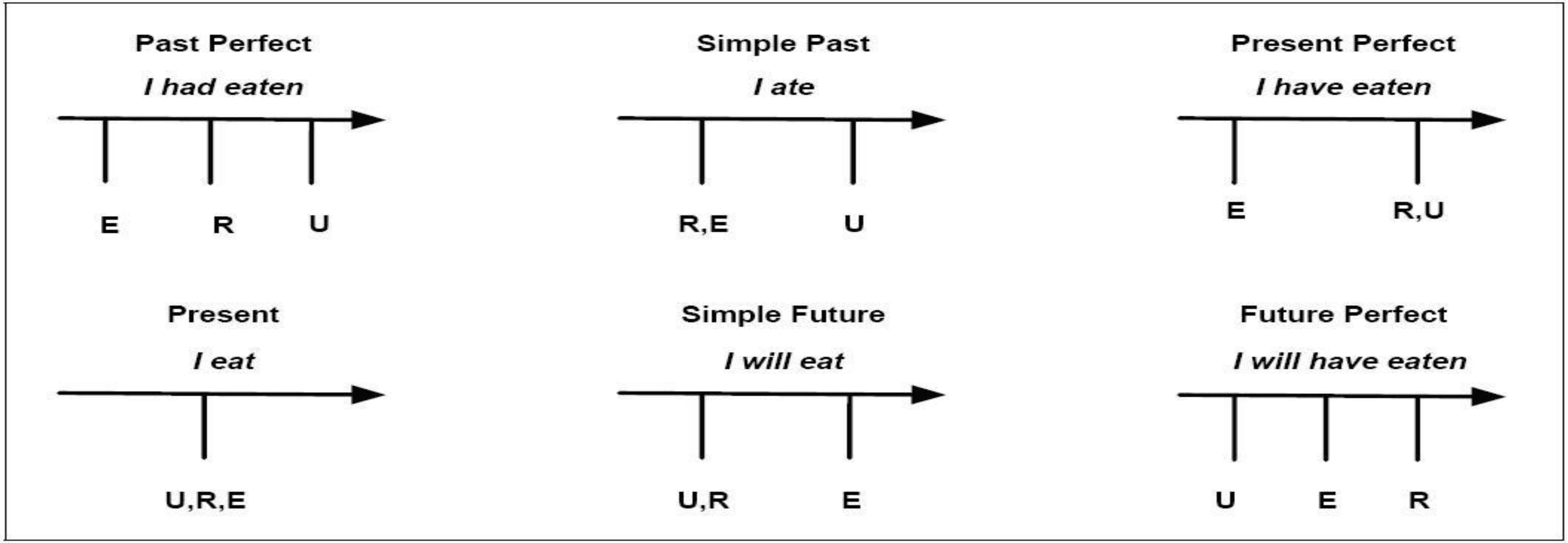
- Reification
  - $\exists$ e: ISA(e,Eating) $\wedge$ Eater(e,Martin) $\wedge$ Eaten(e,Fish)

Example from Jurafsky and Martin

Representing Time

- Example
  - Martin went from the kitchen to the yard
  - ISA(e,Going) ^ Goer(e,Martin) ^ Origin (e,kitchen) ^ Target (e,yard)
- Issue
  - no tense information: past? present? future?
- Fluents
  - A predicate that is true at a given time: T(f,t)

Representing Time



Example from Jurafsky and Martin

Representing time

- ∃ *i,e,w,t: Isa(w,Arriving)* ∧ *Arriver(w,Speaker)* ∧ *Destination(w,NewYork)* ∧ *IntervalOf(w,i)* ∧ *EndPoint(i,e)* ∧ *Precedes (e,Now)*
- ∃ *i,e,w,t: Isa(w,Arriving)* ∧ *Arriver(w,Speaker)* ∧ *Destination(w,NewYork)* ∧ *IntervalOf(w,i)* ∧ *MemberOf(i,Now)*
- ∃ *i,e,w,t: Isa(w,Arriving)* ∧ *Arriver(w,Speaker)* ∧ *Destination(w,NewYork)* ∧ *IntervalOf(w,i)* ∧ *StartPoint(i,s)* ∧ *Precedes (Now,s)*

Example from Jurafsky and Martin

Aspect

- Stative
  - I know my departure gate
- Activity
  - John is flying
    (no particular end point)
- Accomplishment
  - Sally booked her flight
    (natural end point and result in a particular state)
- Achievement
  - She found her gate
- Figuring out statives:
  - I am needing the cheapest fare.
  - I am wanting to go today.
  - Need the cheapest fare!

Example from Jurafsky and Martin

Representing Beliefs

- Example
  - Milo believes that Martin ate fish
- One possible representation
  - $\exists$ e,b: ISA(e,Eating) $\wedge$ Eater(e,Martin) $\wedge$ Eaten(e,Fish) $\wedge$ ISA(b,Believing) $\wedge$ Believer(b,Milo) $\wedge$ Believed(b,e)
- However this implies (by dropping some of the terms) that “Martin ate fish” (without the Belief event)
- Modal logic
  - Possibility, Temporal Logic, Belief Logic

Representing Beliefs

- Want, believe, imagine, know: all introduce hypothetical worlds
- I believe that Mary ate British food.
- Reified example:
  - *∃ u,v: Isa(u,Believing) ∧ Isa(v,Eating) ∧ Believer (u,Speaker) ∧ BelievedProp(u,v) ∧ Eater(v,Mary) ∧ Eaten(v,BritishFood)*

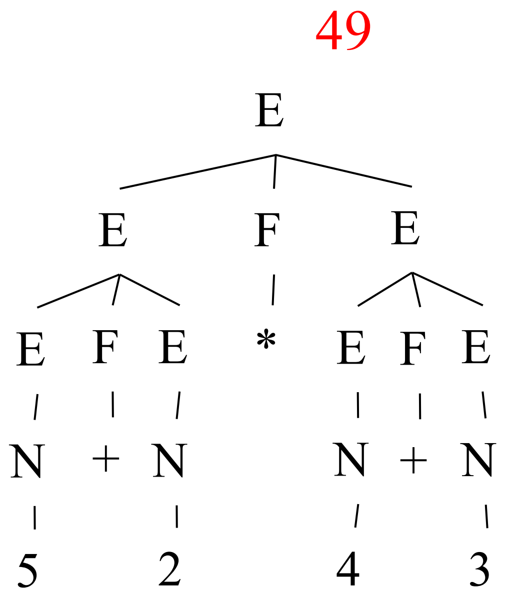
  However this implies also*:*
  - *∃ u,v: Isa(v,Eating) ∧ Eater(v,Mary) ∧ Eaten(v,BritishFood)*
- Modal operators:
  - *Believing(Speaker,Eating(Mary,BritishFood))* - not FOPC! – predicates in FOPC hold between objects, not between relations.
  - *Believes(Speaker, ∃ v: ISA(v,Eating) ∧ Eater(v,Mary) ∧ Eaten(v,BritishFood))*

Introduction to NLP

362.
First Order Logic

Semantics

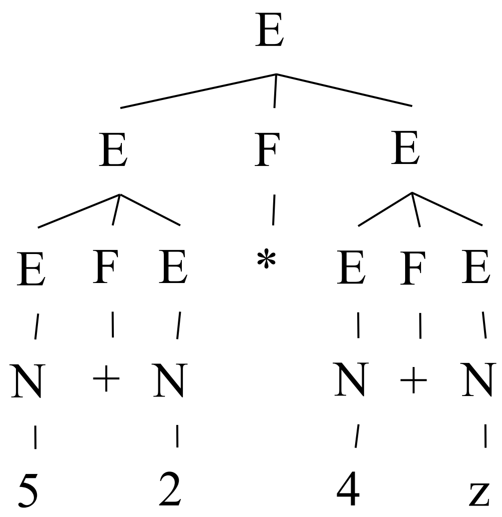
- What is the meaning of: (5+2)*(4+3)?
- Parse tree



Semantics

- What if we had (5+2)*(4+z)?

mult(add(5,2),add(4,z))



What about (English) sentences?

- Socrates is a human.
- Every human is mortal.

Representing Meaning

- Goal
  - Capturing the meaning of linguistic utterances using formal notation
- Linguistic meaning
  - “It is 8 pm”
- Pragmatic meaning
  - “It is time to leave”
- Semantic analysis:
  - Assign each word a meaning
  - Combine the meanings of words into sentences
- *I bought a book*:

$\exists\ x,y$: Buying(x) ^ Buyer($speaker,x$) ^ BoughtItem(y,x) ^ Book(y)

Buying (Buyer=speaker, BoughtItem=book)


LOOK SIMBA, THIS IS LINGUISTICS. YOU MAY STUDY ANY PART OF THE LAND.
WHAT ABOUT THAT DARK SHADOWY PLACE?
THAT IS SEMANTICS. YOU MUST NEVER GO THERE.
zipmeme

First Order Logic

- Used to represent
  - Objects – Martin the cat
  - Relations – Martin and Moses are brothers
  - Functions – Martin’s age

First Order Logic

- *Formula* $\rightarrow$ AtomicFormula | Formula Connective Formula
  | Quantifier Variable Formula | ¬ Formula | (Formula)
- AtomicFormula $\rightarrow$ Predicate (Term…)
- Term $\rightarrow$ Function (Term…) | Constant | Variable
- Connective $\rightarrow$ ∧ | ∨ | ⇒
- Quantifier $\rightarrow$ ∀ | ∃
- Constant $\rightarrow$ M | Martin
- Variable $\rightarrow$ *x* | *y* | …
- Predicate $\rightarrow$ Likes | Eats | …
- Function $\rightarrow$ AgeOf | ColorOf | …

Types

- Base types
  - e (entity) – objects
  - t (truth values)
- Complex types
  - If a is a type and b is a type, then a→b is a type.
  - (a→b)(a)=b
- Example
  - Type of *Mary* = e
  - Type of *sleeps* = e→t
  - Type of *sleeps(Mary)* = t
  - Type of ^ *= t→t*
  - Type of *^(sleeps(Mary))* = t
  - * *^(sleeps)*   - not well typed

Lambda Expressions

- Example
  - inc(x) = λx x+1
  - then inc(4) = (λx x+1)(4) = 5
- Example
  - add(x,y) = λx,λy(x+y)
  - then add(3,4) = (λx,λy(x+y))(3)(4)= (λy 3+y)(4) = 3+4 = 7
- Useful for semantic parsing (see later)

Lambda Expressions

- λx.*like*(x,Mary)
- λx.*like*(Mary,x)
- λx.(λy.*like*(x,y))
- λP.P(Mary)
  - property is true of Mary

Lambda Expressions

- [λx.sleeps(x)](Mary)=sleeps(Mary)
- [λx.likes(Mary,x)](John)=likes(Mary,John)
- [λx.likes(x,y)](Mary)=likes(Mary,Mary)

Example

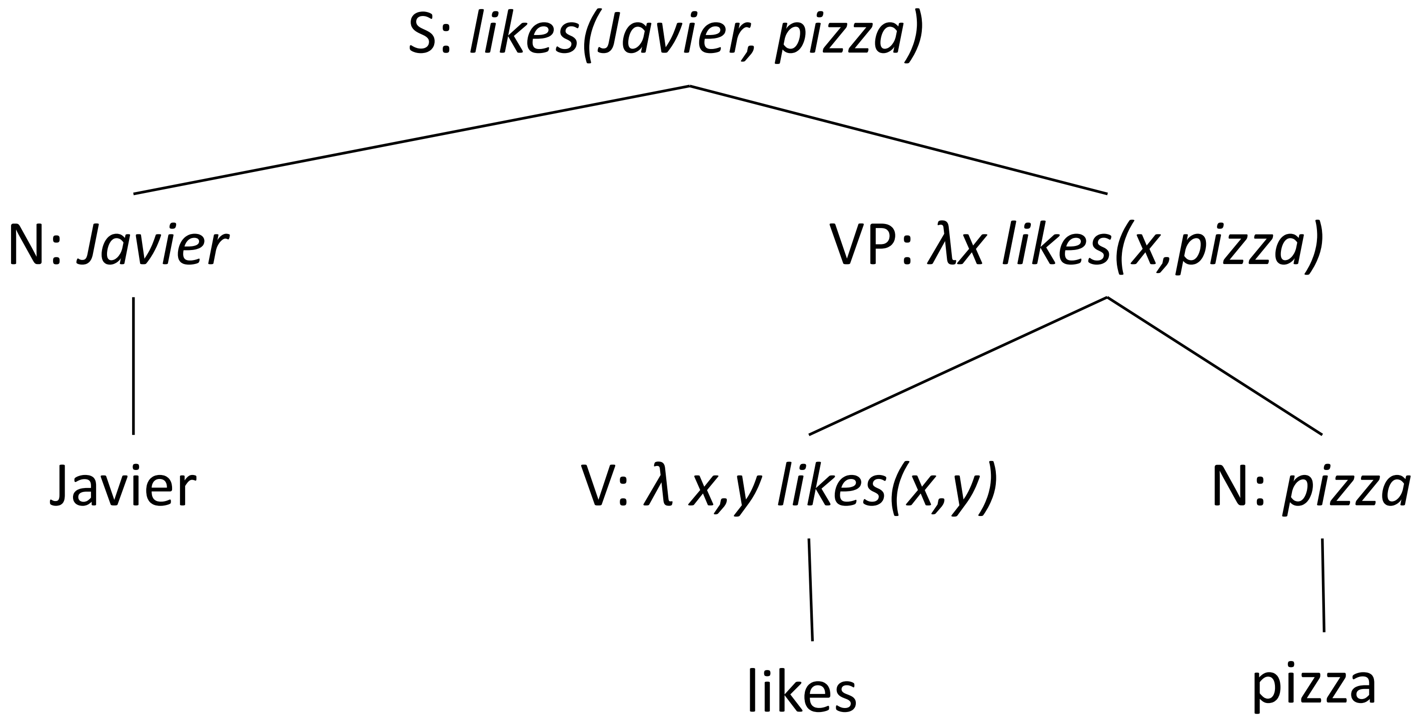
- Input
  - Javier likes pizza
- Output
  - *like(Javier, pizza)*

Example

```
S  -> NP VP    {VP.Sem(NP.Sem)}     t
VP -> V NP     {V.Sem(NP.Sem)}      <e,t>
NP -> N        {N.Sem}              e
V  -> likes    {λ x,y likes(x,y)    <e,<e,t>>
N  -> Javier   {Javier}             e
N  -> pizza    {pizza}              e
```

Semantic Parsing (preview)

- Associate a semantic expression with each node



Introduction to NLP

364.

Inference

Modus Ponens

- Modus ponens:

$$\frac{\alpha \qquad \alpha \Rightarrow \beta}{\beta}$$

- Example:

$$\frac{\mathit{Cat(Martin)} \qquad \forall x: \mathit{Cat}(x) \Rightarrow \mathit{EatsFish}(x)}{\mathit{EatsFish(Martin)}}$$

Inference

- Forward chaining
  - as individual facts are added to the database, all derived inferences are generated
- Backward chaining
  - starts from queries
  - Example: the Prolog programming language
- Prolog example
  - father(X, Y) :- parent(X, Y), male(X).
    parent(john, bill).
    parent(jane, bill).
    female(jane).
    male (john).
    ?- father(M, bill).

The Kinship Domain

- Brothers are siblings

  $\forall$x,y *Brother(x,y)* $\Rightarrow$ *Sibling(x,y)*

- One's mother is one's female parent

  $\forall$m,c *Mother(c)* = m $\Leftrightarrow$ *(Female(m)* $\wedge$ *Parent(m,c))*

- “Sibling” is symmetric

  $\forall$x,y *Sibling(x,y)* $\Leftrightarrow$ *Sibling(y,x)*

Universal Instantiation

- Every instantiation of a universally quantified sentence is entailed by it:

$$\frac{\forall v\ \alpha}{\text{Subst}(\{v/g\}, \alpha)}$$

for any variable *v* and ground term *g*

- E.g., $\forall x\ Cat(x) \wedge Fish(y) \Rightarrow Eats(x,y)$ yields:

  $Cat(Martin) \wedge \text{Fish}(\text{Blub}) \Rightarrow Eats(Martin,Blub)$

Existential Instantiation

- For any sentence α, variable *v*, and constant symbol *k* that does not appear elsewhere in the knowledge base:

$$\frac{\exists v\ \alpha}{\mathrm{Subst}(\{v/k\}, \alpha)}$$

- E.g., $\exists x\ Cat(x) \wedge EatsFish(x)$ yields:

$$Cat(C_1) \wedge EatsFish(C_1)$$

provided C_1 is a new constant symbol, called a Skolem constant

Unification

- If a substitution θ is available, unification is possible
- Examples:
  - p=Eats(x,y), q=Eats(x,Blub), possible if θ = {y/Blub}
  - p=Eats(Martin,y), q=Eats(x,Blub), possible if θ = {x/Martin,y/Blub}
  - p=Eats(Martin,y), q=Eats(y,Blub), fails because Martin$\neq$Blub
- Subsumption
  - Unification works not only when two things are the same but also when one of them subsumes the other one
  - Example: All cats eat fish, Martin is a cat, Blub is a fish