Applications for NLP
Lecture 5: Linguistics in NLP - 4

Sai Qian

Email: sai.qian@loria.fr
Semagramme, LORIA & INRIA Nancy - Grand Est

September 29, 2011

Outline

1. Review

2. More Semantics
   - Type Theory & Lambda Calculus
   - Dynamic Semantics
   - Lexical Semantics
   - Semantics in NLP

In The Last Lecture

- Semantics is all about "**meaning**"
  - Lexical Meaning - Lexical Semantics
  - Sentence Meaning - Compositional Semantics
  - Discourse Meaning - Discourse Semantics
- Predicate Logic & First Order Logic, some confusing pairs:
  - Constants and Variables
  - Terms and Formulae
  - Functions and Predicates

Type Theory

- Church's **Theory of Types**: developed by Alonzo Church, father of **lambda calculus**
- Montague's semantic framework was based on Church's type theory
- Informal Definition (recursive)
  - There is a set of **basic types** $\{t_1, t_2 ... t_n\}$
  - If x and y are types, then $x \rightarrow y$[1] is also a type, we call it **complex type**
- In Montague Semantics, two basic types e and t
  - e denotes the type of entities (or individuals)
  - t denotes the type of propositions (or truth values)
  - Other type examples: $e \rightarrow t$, $(e \rightarrow t) \rightarrow t$, $(e \rightarrow t) \rightarrow (e \rightarrow t)$...

[1] Sometimes also denoted as $< x, y >$.

Lambda Calculus & Lambda Term

- Lambda Calculus can be viewed as an **extension** of FOL
- λ expressions
  - General Form: $\lambda VAR.\phi$
    VAR stands for **variables**, ϕ stands for **formulas** (not term)
  - Examples: $\lambda x.P(x)$, $\lambda y.\phi$, $\lambda x.man(x)$
- **Bound/Free** variable: depending on whether VAR appears in the **scope** of the λ operator or not **in the λ term**

Example (Bound/Free Variables)

Indicate all bound and free variables (if there is any) in the following λ expression:

1. $\lambda x.\lambda y.(P(x) \wedge Q(y)) \Rightarrow x, y$ Bound
2. $\lambda x.\lambda y.(P(x) \wedge Q(y) \wedge M(z)) \Rightarrow x, y$ Bound, z Free
3. $\lambda x.\lambda y.P(x) \wedge Q(y) \Rightarrow x$ Bound, y Free

α-conversion & β-reduction

- α-conversion: the **renaming** of bound variables in a λ expression, yielding an equivalent expression
  - $\lambda x.P(x) \Rightarrow_\alpha \lambda y.P(y)$
  - $\lambda x.\lambda y.(P(x) \wedge Q(y)) \Rightarrow_\alpha \lambda a.\lambda b.(P(a) \wedge Q(b))$
- β-reduction: the process that the corresponding variable in the formula is **rewritten** by the argument, until the function itself is reduced to a simpler form
  - $\lambda x.P(x)@y \Rightarrow_\beta P(y)$
  - $\lambda x.run(x)@J \Rightarrow_\beta run(J)$
  - $\lambda x.\lambda y.(P(x) \wedge Q(y))@a@b \Rightarrow_\beta$
    $\lambda y.(P(a) \wedge Q(y))@b \Rightarrow_\beta$
    $P(a) \wedge Q(b)$

How To Do Lambda Calculus?

Steps of doing Lambda Calculus:

1. Determine which expression is the **function** and which is the **argument**
2. Apply the argument to the function
3. β-**reduce** the conjoined element

Example (λ Calculus)

$\lambda P.\lambda x.(P(x) \wedge good(x))@\lambda x.man(x) \Rightarrow_\alpha$
$\lambda P.\lambda x.(P(x) \wedge good(x))@\lambda y.man(y) \Rightarrow_\beta$
$\lambda x.(\lambda y.man(y)@x \wedge good(x)) \Rightarrow_\beta$
$\lambda x.(\lambda y.man(y)@x \wedge good(x)) \Rightarrow_\beta$
$\lambda x.(man(x) \wedge good(x))$

Semantics Exercise

Question

Compute the result for the following Lambda Calculus:

- $\lambda x.x@y$
- $\lambda x.y@y$
- $\lambda x.P(x) \wedge Q(y)@y$
- $\lambda x.y.P(x) \wedge Q(y)@y@z$
- $\lambda x.xy@(\lambda z.zy)$
- $\lambda x.xx@\lambda x.xx$

Semantics Exercise

Answer

Compute the result for the following Lambda Calculus:

- $\lambda x.x@y \Rightarrow_{\alpha,\beta} y$
- $\lambda x.y@y \Rightarrow_{\alpha,\beta} y$
- $\lambda x.P(x) \wedge Q(y)@y \Rightarrow_{\alpha,\beta} P(y) \wedge Q(z)$
- $\lambda x.y.P(x) \wedge Q(y)@y@z \Rightarrow_{\alpha,\beta} P(y) \wedge Q(z)$
- $\lambda x.xy@(\lambda z.zy) \Rightarrow_{\alpha,\beta} yx$
- $\lambda x.xx@\lambda x.xx \Rightarrow_{\alpha,\beta} \lambda x.xx$

Typed Lambda Calculus

- Definition: in **Typed Lambda Calculus**, everything (variable, constant or predicate) in the λ expression has its type
- For a **predicate** relation (function-argument), types are strictly restricted
  - The type of the predicate and the type of the argument(s) **must** match
  - Example: $xy/x@y \to y(t_1), x(t_1 \to t_2), xy/x@y(t_2)$
  - As a result, "$\lambda x.xx$" is prohibited in Typed Lambda Calculus, the formula "xx" is not typeable

Types & Syntactic Categories

Every syntactic category has its own type

- Common Nouns
  - Example: $\lambda x.man(x)$, $\lambda x.car(x)$
  - Type: $e \rightarrow t$
  - The **property** of being an x such that x is a man
- Determiners
  - Example: $\lambda PQ.\exists x.(P(x) \wedge Q(x))$, $\lambda PQ.\forall x.(P(x) \rightarrow Q(x))$
  - Type: $(e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t))$
- Noun Phrases
  - Example: $\lambda Q.\exists x.(man(x) \wedge Q(x))$, $\lambda Q.\forall x.(man(x) \rightarrow Q(x))$
  - Type: $(e \rightarrow t) \rightarrow t$

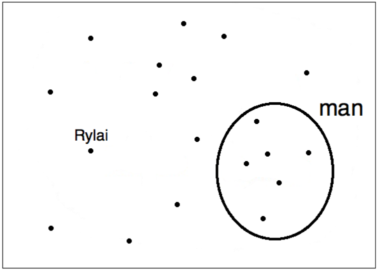
Types & Syntactic Categories

- Proper Names
  - Example: $RYLAI$, $LINA$ or $\lambda P.P(RYLAI)$, $\lambda P.P(LINA)$
  - Type: e or $(e \rightarrow t) \rightarrow t$
- Intransitive Verb (similar to Common Noun)
  - Example: $\lambda x.run(x)$, $\lambda x.sleep(x)$
  - Type: $e \rightarrow t$
- Transitive Verb
  - Example: $\lambda OS.S(\lambda x.(O\lambda y.love(x, y)))$
  - Type: $((e \rightarrow t) \rightarrow t) \rightarrow (((e \rightarrow t) \rightarrow t) \rightarrow t)$
- Adjective[2]
  - Example: $\lambda Px.(P(x) \wedge red(x))$
  - Type: $(e \rightarrow t) \rightarrow (e \rightarrow t)$

[2] The λ expression for adjectives is a complicated problem, here I just give a simple example.

Some Deeper Questions About Types

- Why common nouns are of type "$e \rightarrow t$"? ($\lambda x.man(x)$)
  - In natural language, common nouns are properties, or sets
  - $\lambda x.man(x)$ denotes the set of "man" or the properties that every "man" shares in common
- How about Proper Names?
  - Take it as a constant ($RYLAI$)
  - $(e \rightarrow t) \rightarrow t$: set of sets, or set of properties
  - The properties that the entity possesses ($\lambda P.P(RYLAI)$)
- NPs? Intransitive Verbs?



Compositionality

Frege's Principle (Compositionality)

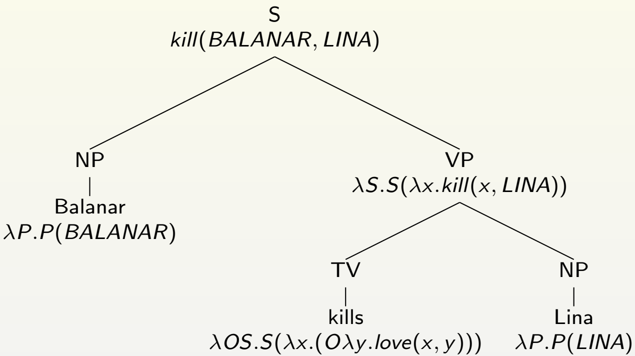
The meaning of a complex expression is determined by the meanings of its constituents and the structure they are combined.

Montague Semantics

Type Theory + Lambda Calculus + First Order Logic (FOL) + Compositionality

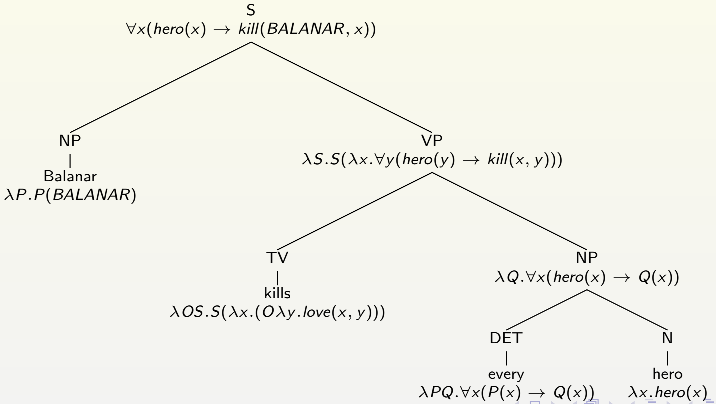
Doing Semantics Compositionally

Balanar kills Lina.



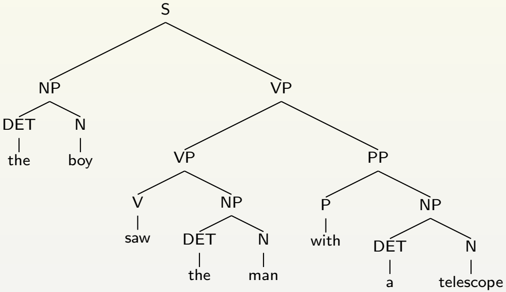
Doing Semantics Compositionally

Balanar kills every hero.



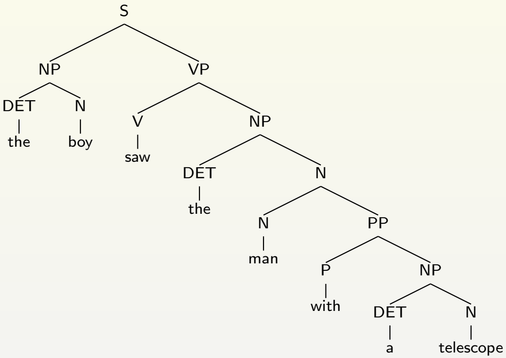
Syntactic Ambiguity

- Definition: the same sequence of words is interpreted as having different **syntactic structures**
- Examples
  - *The boy saw the man with a telescope.*



Syntactic Ambiguity Continued

- Examples
  - *The boy saw the man with a telescope.*



Semantic Ambiguity

- Definition: with the same **syntactic structure**, there exists different **meaning interpretations**
- *Every man loves a woman.*
  1. There is a celebrity woman, and every man in the world likes her
  2. For every man in the world, there is a woman that he likes
- Meaning representation for the two readings
  1. $\exists y(woman(y) \wedge \forall x(man(x) \rightarrow love(x, y)))$
  2. $\forall x(man(x) \rightarrow \exists y(woman(y) \wedge love(x, y)))$
- How to resolve scope ambiguity?
  - Two alternative syntactic structures[3]
  - Cooper Storage: computationally expensive

[3] This solution is not general enough because it is not practical to assign every sentence two structures

Where Compositional FOL Fails?

1. *A hero dies. He revives at the fountain.*
   - What is the FOL translation?
     - $\exists x(hero(x) \wedge die(x)) + revive(x)$???
     - $\exists x(hero(x) \wedge die(x) \wedge revive(x))$
   - Scope of "$\exists x$" is extended, **not** a systematic solution
2. *Every farmer who owns a donkey beats it.*
   - What is the FOL translation of this sentence?
     - $\forall x(farmer(x) \wedge \exists y(donkey(y) \wedge own(x, y)) \rightarrow beat(x, y))$???
     - $\forall x \exists y(farmer(x) \wedge donkey(y) \wedge own(x, y) \rightarrow beat(x, y))$???
     - $\forall x \forall y(farmer(x) \wedge donkey(y) \wedge own(x, y) \rightarrow beat(x, y))$
   - Same meaning as "*if a farmer owns a donkey, then he beats it.*"
   - Why **existential** quantifier "*a*" should be translated into **universal** quantifier?

Dynamic Semantics

- **Dynamic Semantics**
  - Why dynamic - pieces of text or discourse are viewed as instructions to **update** an existing context with new information
  - In a slogan - **meaning is of context change potential**
- Other dynamic semantic formalisms
  - **File Change Semantics** by Irene Heim
  - **Dynamic Predicate Logic** by Groenendijk & Stokhof
  - **Dynamic Treatment to MS** by Philippe de Groote[4]

[4] More information refers to the paper "Towards a Montagovian account of dynamics"

Motivation for "**Event**"

In Davidson's 1967 paper "The Logical Form of Action Sentences", **event** was described as a **linguistic entity** in action sentences

- Modifiers
    - Examples:
        1. *Brutus stabbed Caesar in the back with a knife.*
        2. *Brutus stabbed Caesar in the back.*
        3. *Brutus stabbed Caesar with a knife.*
        4. *Brutus stabbed Caesar.*
    - How to represent meanings of the 4 sentence in a logical way, while pertaining the entailment relation among them?
        - Higher Order Modifiers
        - Multiple Predicates: $stab_1$, $stab_2$, $stab_3$
        - A "Complete" Predicate: $stab(subj, obj, location, tool \ldots)$
        - **Event**

Motivation for "**Event**" Continued

- Modifiers
  - Representation with Event
    1. $\exists e(stab(e) \wedge subj(e, B) \wedge obj(e, C) \wedge in(e, back) \wedge with(e, knife))$
    2. $\exists e(stab(e) \wedge subj(e, B) \wedge obj(e, C) \wedge in(e, back))$
    3. $\exists e(stab(e) \wedge subj(e, B) \wedge obj(e, C) \wedge with(e, knife))$
    4. $\exists e(stab(e) \wedge subj(e, B) \wedge obj(e, C))$
  - Implication Result: $S_1 \rightarrow S_2 \rightarrow S_4$, $S_1 \rightarrow S_3 \rightarrow S_4$
- Vague Semantic Ambiguities
  - Examples:
    - *John and Mary went to school.*
    - *John loves all women he meets.*
  - How many "events" are there?
- "Event" Everywhere
  - *I'll sleep after that.*
  - *I'll do that first.*

Verb Classification 1

- Intuitional Way
  - **Main Verb** (Action Verb): a verb that is used to describe an action or an event
    - Intransitive verb, transitive verb, di-transitive verb...
    - Example: *run, play, give...*
  - **Auxiliary Verb**: a verb that does not have a real meaning by itself, while it requires to go along with another action verb
    - Expressing tense, passiveness, modality
    - *Examples: have, be, could, should...*
  - **Linking Verb** (State Verb): a verb that denotes the state of the object
    - Has meaning, but can not describe an action or event
    - Example: *seem, smell, be...*

Verb Classification 2

- Vendler's Way[5]
  - **State Verb**: no effect of meaning changing or modifying during the time span
    - Example: *John loves Mary. John is tall.*
  - **Activity Verb**: describing a concrete on-going action, that has internal change and duration, but end point is not necessary
    - Example: *John runs. John walks along the river.*
  - **Achievement Verb**: besides describing an event or action, an end point or culmination is required, the event should be without duration
    - Example: *Mary arrived at the destination. John reaches the top of the mountain*
  - **Accomplishment Verb**: nearly the same as the achievement verb except that the described event needs to have duration
    - Example: *John consumed an apple.*

[5] This classification is based on Vendler's 1967 paper "Verbs and Times".

Thematic Roles

- What is “Thematic Role”?
  - View the verb as the **center** of the sentence, the **roles** that the rest parts of the sentence play
  - Coarse Example: subject, object, location...
- Why we need “Thematic Role”?
  - Assumption: languages do not differ in expressive power
  - A **universal representation** at the semantic level
- Examples
  - *Brutus (**Agent**) stabbed Caesar (**Theme**) with a knife (**Instrument**).*
  - *Caesar (**Theme**) was stabbed by Brutus (**Agent**) with a knife (**Instrument**).*
- Thematic Roles do not change with sentence structures

Thematic Roles Details

- It is still **controversial** to declare how many thematic roles are there
- A general framework

Thematic Role	Syntactic Correspondence
Agent	Subject
Theme	Direct object; subject of "be"
Goal	Indirect object, or with "to"
Benefactive	Indirect object, or with "for"
Instrument	Object of "with"; subject
Experiencer	Subject

Semantics in NLP - WSD

- Word Sense Disambiguation
  - A word could have **several** meanings
  - Disambiguation between different meanings is necessary
  - Needed for most NLP that involve semantics
  - Selectional restrictions to identify meanings intended in given context
    1. *The astronomer saw* **the star***.*
    2. *The astronomer married* **the star***.*
  - Statistical evidence derived from large corpora
    1. *John sat on* **the bank***.*
    2. *John went to* **the bank***.*
    3. *King Kong sat on* **the bank***.*

Semantics in NLP -LR

- Lexical Relations
  - **Relations** among word meanings are also very important for natural language based applications
  - The most commonly used lexical relations
    - **Hyponymy** (is_a) e.g., *dog* is a hyponym of *animal*, *animal* is the hypernym of *dog*
    - **Meronymy** (part-of) e.g., *arm* is a meronym of *body*
    - **Synonymy** e.g., *eggplant* & *aubergine*, *fall* & *autumn*
    - **Antonymy** e.g., *big* & *little*, *tall* & *short*
  - **WordNet**: a good source for lexical relations

Semantics in NLP - LR Continued

- In natural language applications, the most commonly used lexical relation is hyponymy such that
  - **Semantic Classification** (e.g., selectional restrictions, named entity recognition)
  - **Shallow Inference** (e.g., "*X murdered Y*" implies "*X killed Y*")
  - **Word Sense Disambiguation**
  - **Machine Translation** (if a term cannot be translated, substitute a hypernym)

Semantics in NLP - Other

- **Shallow** Semantic Parsing (Role Labeling)
  - Based on the event structure, labeling the thematic roles of each components
  - Potential Situations
    - Historical document
    - Newspaper, live sport commentary (football, basketball)
- Natural Language **Understanding**
  - Ultimate goal for NLP
  - Same words, different meanings
    1. *John loves Mary.* $\Rightarrow love(J, M)$
    2. *Mary loves John.* $\Rightarrow love(M, J)$

Summary

- Type Theory & Lambda Calculus
- Typed Lambda Calculus and natural language semantics
- Compositional Semantics - obtaining meaning representation in a systematic way
- Dynamic Semantics
- Event Semantics & Thematic Roles
- Semantics in NLP applications