First Order Predicate Calculus

By:
B.Senthil Kumar
Asst.Prof / CSE
Natural Language Processing

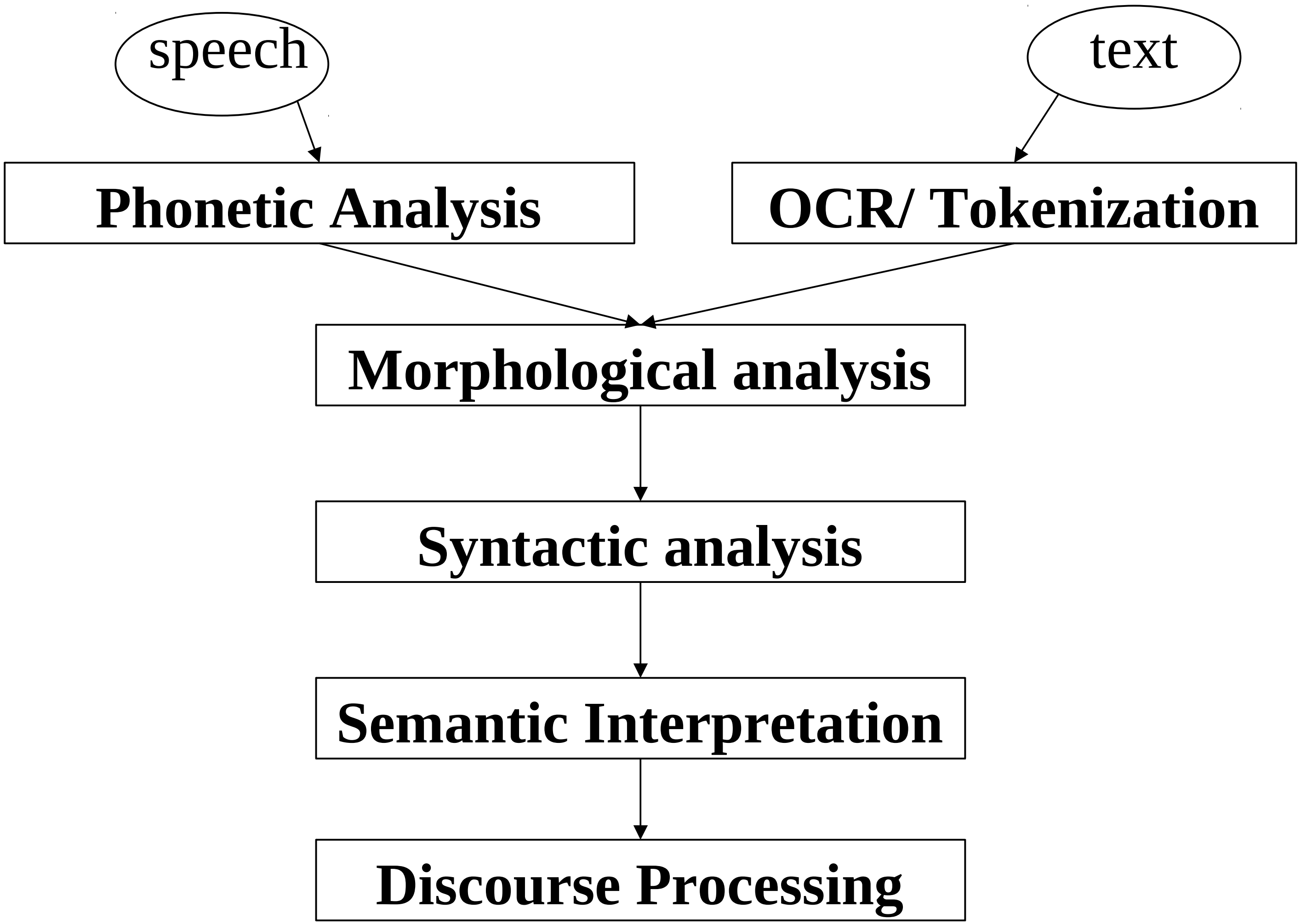
Slides from:

* Speech and Language Processing, Jurafsky and Martin
* Husni Al-Muhtaseb
* Ching-Long Yeh
* Heshaam Feili
* Kathy McCoy

Agenda

- First Order Predicate Calculus
- Linguistically Relevant Concepts

NLP Pipeline



First Order Predicate Calculus

- FOPC is a flexible, well-understood, and computationally tractable approach to the representation of knowledge.
- It provides a strong computational basis for the verifiability, inference, and expressiveness requirement.
- The most attractive feature of FOPC:
    - It makes very few specific commitments as to how things ought to be represented.
    - The represented world consists of objects, properties of objects, and relations among objects.

First Order Predicate Calculus

$$
\begin{array}{rcl}
\textit{Formula} & \rightarrow & \textit{AtomicFormula} \\
& | & \textit{Formula Connective Formula} \\
& | & \textit{Quantifier Variable}, \ldots \textit{ Formula} \\
& | & \neg \textit{ Formula} \\
& | & (\textit{Formula}) \\
\\
\textit{AtomicFormula} & \rightarrow & \textit{Predicate}(\textit{Term}, \ldots) \\
\\
\textit{Term} & \rightarrow & \textit{Function}(\textit{Term}, \ldots) \\
& | & \textit{Constant} \\
& | & \textit{Variable} \\
\\
\textit{Connective} & \rightarrow & \wedge \mid \vee \mid \Rightarrow \\
\textit{Quantifier} & \rightarrow & \forall \mid \exists \\
\textit{Constant} & \rightarrow & A \mid \textit{VegetarianFood} \mid \textit{Maharani} \cdots \\
\textit{Variable} & \rightarrow & x \mid y \mid \cdots \\
\textit{Predicate} & \rightarrow & \textit{Serves} \mid \textit{Near} \mid \cdots \\
\textit{Function} & \rightarrow & \textit{LocationOf} \mid \textit{CuisineOf} \mid \cdots
\end{array}
$$

Elements of FOPC

- **Term –** the FOPC device for representing objects.
- Three basic building blocks: constants, functions, variables
  - **Constants**
    - Specific objects in the world being described.
    - Depicted as single capitalized letters *A*, *B*, or proper nouns such as *Maharani*, *Ram*
  - **Functions**
    - Concepts that are expressed in English as genitives,
      - *the location of Maharani*, or *Maharani's location*
      - *LocationOf* (*Maharani*)
    - Refer to unique objects, though appearing similarly as predicates

Elements of FOPC

- Three basic building blocks: constants, functions, variables
    - **Variables**
        - Depicted as single lower-case letters
        - Ability to make assertions and draw inferences about objects without having to make reference to any particular named object

Elements of FOPC

- **Predicates –** Relations that hold among objects.
  - Predicates are symbols refer to, or name, the relations that hold among some fixed number of objects in a given domain
    - *Serves*(*Maharani*,*VegetarianFood*) – a two-place predicate
    - *Restaurant*(*Maharani*) – a one-place predicate
  - Complex formula, through the use of **logical connectives**
  - (14.17) *I only have five dollars and I don't have a lot of time.*
    - $Have(Speaker, FiveDollars) \wedge \neg Have(Speaker, LotOfTime)$

The Semantics of FOPC

- How various objects, properties, and relations presented on a FOPC acquire their meanings?

  by virtue of their correspondence to objects, properties, and relations out in the external world being modeled by the knowledge base
- FOPC sentences can therefore be assigned a value of *True* or *False*
- The interpretations of formulas involving logical connectives is based on the meaning of the components in the formulas combined with the meaning of connectives they contain.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$
False	*False*	*True*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*	*True*
True	*False*	*False*	*False*	*True*	*False*
True	*True*	*False*	*True*	*True*	*True*

Variables and Quantifiers

- Variables are used in two ways in FOPC:
    - To refer to particular anonymous objects and
    - To refer generically to all objects in a collection
- The two uses are made possible through the use of **quantifiers**.
    - The two operators are the existential quantifier - ∃ ("*there exists*") and the universal quantifier - ∀ ("*for all*")
    - The need for existential quantifier is due to the presence of an indefinite noun phrase

(14.19) *a restaurant that serves Mexican food near ICSI*

$\exists x\ Restaurant(x)$
$\quad \wedge\ Serves\ (x, MexicanFood)$
$\quad \wedge\ Near\ (LocationOf(x), LocationOf(ICSI))$

Variables and Quantifiers

- For this sentence to be true there must be at least one object such that if substituted for *x*, the resulting sentence would be true
- If *Gateway* is a Mexican restaurant near ICSI, then:
  - Substituting for *x* results in:

$$\begin{aligned}&Restaurant(Gateway)\\&\quad\wedge\ Serves\ (Gateway, MexicanFood)\\&\quad\wedge\ Near\ (LocationOf(Gateway), LocationOf(ICSI))\end{aligned}$$

  - The sentence will be *true* if all of its three atomic formulas are *true*.

Variables and Quantifiers

- $\forall$ operator states that for the formula to be true, the substitution of <u>*any* object in the knowledge base</u> for the universally quantified variable should result in a true formula.
- Consider the following example:
  - (14.20) All vegetarian restaurants serve vegetarian food.

    $$\forall x\ VegetarianRestaurant(x) \Rightarrow Serves(x, VegetarianFood)$$

  - Case 1: Set of objects consisting of vegetarian restaurants:

    $$VegetarianRestaurant(Maharani)$$
    $$\Rightarrow\ Serves(Maharani, VegetarianFood)$$

  - If consequent is true or both antecedent and the consequent have the value *True*, then the sentence itself is *True*.

Variables and Quantifiers

- Consider the following example:
  - (14.20) All vegetarian restaurants serve vegetarian food.

    $\forall x\ VegetarianRestaurant(x) \Rightarrow Serves(x, VegetarianFood)$

  - Case 2: Set of a objects that are not vegetarian restaurants:

    $VegetarianRestaurant(Gateway)$

    $\Rightarrow\ Serves(Gateway, VegetarianFood)$

  - Since the antecedent of the implication is *False*, the sentence is always *True* satisfying the $\forall$ constraint.
  - There is no restrictions on objects that can be substituted for x by this kind of reasoning.

Inference

- Inference
  - The ability to add valid new propositions to a knowledge base, or
  - To determine the truth of propositions not explicitly contained within a knowledge base.

– **Modus ponens –** inference method provided by FOPC.

– If the left-hand side of an implication rule is present in the knowledge base, then the right-hand side of the rule can be inferred.

$$\frac{\begin{array}{l}\alpha \\ \alpha \Rightarrow \beta\end{array}}{\beta} \qquad \frac{\begin{array}{l}\mathit{VegetarianRestaurant}(\mathit{Rudys}) \\ \forall x\ \mathit{VegetarianRestaurant}(x) \Rightarrow \mathit{Serves}(x, \mathit{VegetarianFood})\end{array}}{\mathit{Serves}(\mathit{Rudys}, \mathit{VegetarianFood})}$$

– The formula *VegetarianRestaurant*(*Rudys*) matches the antecedent thus using modus ponen concludes *Serves*(*Rudys*, *VegetarianFood*)

Inference

- Modus ponens used in two ways:
- Forward chaining:
  - As soon as new fact is added to **kb** all applicable implication rules are found and applied, each resulting in addition of new facts to the **kb**.
  - All inference is preformed in advance, hence facts will be present always.
- Backward chaining:
  - Modus ponen run in reverse.
    - Check if the query formula is present in the kb
    - If not, search for applicable implication rule (consequent matches the query) present in kb
    - Query is proved if any of the antecedent is shown to be true

Inference

- Backward chaining:
  - **Prolog** programming language is a backward chaining system.

$VegetarianRestaurant(Rudys)$

$\forall x\ VegetarianRestaurant(x) \Rightarrow Serves(x, VegetarianFood)$

? $Serves(Rudys, VegetarianFood)$.

True.

after substituting the constant (*Rudys*) for variable *(x)*, prove the antecedent of the rule

Linguistically Relevant Concepts

- Categories
- Events
- Representing Time
- Aspect
- Representing Beliefs

Categories

- The most common way to represent categories is to create a unary predicate for each category of interest.

  VegetarianRestaurant(*Maharani*)

- Using this method, it is difficult to make assertions about categories themselves.

  MostPopular(*Maharani*, *VegetarianRestaurant*)

  - Not a legal FOPC formula since the arguments to predicates must be *Terms*, not other predicates.

- Solution: **reification**
  - To represent all concepts as full-fledged objects

  ISA(*Maharani*, *VegetarianRestaurant*) – relation between objects and category

  AKO(*VegetarianRestaurant*, *Restaurant*) – relation between categories

Events

- Events consists of single predicates with arguments needed to incorporate all the roles associated:

  Reservation(*Hearer*, *Maharani*, *Today*, *8PM*, *2*)

– Problems
  - Determining the correct number of roles for any given event
  - Representing facts about the roles associated with an event
  - Ensuring that all the <u>correct inferences</u> can be derived directly from the representation of an event
  - Ensuring that <u>no incorrect inferences</u> can be derived from the representation of an event

Events

- Create as many predicates for each of the subcategorization frames

(14.22) I ate.	$Eating_1(Speaker)$
(14.23) I ate a turkey sandwich.	$Eating_2(Speaker, TurkeySandwich)$
(14.24) I ate a turkey sandwich at my desk.	$Eating_3(Speaker, TurkeySandwich, Desk)$
(14.25) I ate at my desk.	$Eating_4(Speaker, Desk)$
(14.26) I ate lunch.	$Eating_5(Speaker, Lunch)$
(14.27) I ate a turkey sandwich for lunch.	$Eating_6(Speaker, TurkeySandwich, Lunch)$
(14.28) I ate a turkey sandwich for lunch at my desk.	$Eating_7(Speaker, TurkeySandwich, Lunch, Desk)$

- Problem
  - High cost, logical relations among events are missing
- Solution: **Meaning postulates**

$$\forall w, x, y, z\ Eating_7(w, x, y, z) \Rightarrow Eating_6\ (w, x, y)$$

  - Suitable for small domains and has scalability problems

Events

- $\exists w, x, y\ Eating(Speaker, w, x, y)$
  $\exists\ w, x\ Eating(Speaker, TurkeySandwich, w, x)$
  $\exists\ w\ Eating(Speaker, TurkeySandwich, w, Desk)$
  $\exists\ w, x\ Eating(Speaker, w, x, Desk)$
  $\exists\ w, x\ Eating(Speaker, w, Lunch, x)$
  $\exists\ w\ Eating(Speaker, TurkeySandwich, Lunch, w)$
  $Eating(Speaker, TurkeySandwich, Lunch, Desk)$
- Advantages:
  - logical connections among formulas without meaning postulates
- Deficiencies:
  - Too many commitments – eating events are associated with a meal
  - Does not let us individuate events

Events

$\exists\, w, x\ Eating(Speaker, w, x, Desk)$……….(a) I ate at my desk

$\exists\, w, x\ Eating(Speaker, w, Lunch, x)$……....(b) I ate lunch

$\exists\, w, x\ Eating(Speaker, w, Lunch, Desk)$…..(c) I ate lunch at my desk

- Given the independent facts a and b, it does not conclude c, using the current representation.
- Employ **reification** to <u>elevate events to objects</u> that can be quantified and related to another objects via sets of defined relations.

$$\exists w\ ISA(w, Eating) \wedge Eater(w, Speaker) \wedge Eaten(w, TurkeySandwich) \qquad -14.23$$

$$\exists w\ ISA(w, Eating) \wedge Eater(w, Speaker) \qquad -14.22$$

$$\exists w\ ISA(w, Eating) \wedge Eater(w, Speaker) \wedge Eaten(w, TurkeySandwich) \wedge MealEaten(w, Lunch) \qquad -14.27$$

Events

- Under reified-event approach:
  - No need to specify a fixed number of arguments – roles can be glued on as appear in the input
  - Roles are not postulated
  - Logical connections among related examples is satisfied without the need for meaning postulates

Representing Time

- The representation of time information in a useful form is the domain of **temporal logic**
- Time flows inexorably forward, and that events are associated with either points or intervals in time, as on a timeline
- An ordering can be imposed on distinct events – one event *precedes* another, if the flow of time leads from the first event to the second

Representing Time

(14.29) I arrived in New York.
(14.30) I am arriving in New York.
(14.31) I will arrive in New York.

$\exists\, w\; ISA(w, Arriving) \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork)$

Add temporal variables representing:
a) the interval corresponding to the event - *IntervalOf*(*w*, *i*)

b) the end point of the event - *EndPoint*(*i*,*e*)

c) temporal predicates relating this end point to the current time - *Precedes*(*e*, *Now*)

* *Precedes* represents the notion that the first time point argument precedes the second in time.

* *Now* refers to the current time.

Representing Time

(14.29) I arrived in New York.

$\exists\, i, e, w, t\ ISA(w, Arriving) \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork)$
$\wedge\ IntervalOf(w, i) \wedge EndPoint(i,e) \wedge Precedes(e, Now)$

Past event

(14.30) I am arriving in New York.

$\exists\, i, e, w, t\ ISA(w, Arriving) \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork)$
$\wedge\ IntervalOf(w, i) \wedge MemberOf(i, Now)$

Present event

(14.31) I will arrive in New York.

$\exists\, i, e, w, t\ ISA(w, Arriving) \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork)$
$\wedge\ IntervalOf(w, i) \wedge EndPoint(i,e) \wedge Precedes(Now, e)$

Future event

Representing Time

- Reichenbach's approach:
- The current moment in time is **equated** with the time of utterance, and is used as a **reference point** for when the event occurred (*before* / *at* / *after*)
- The reference point is separated from utterance time and the event time
- Consider the following:

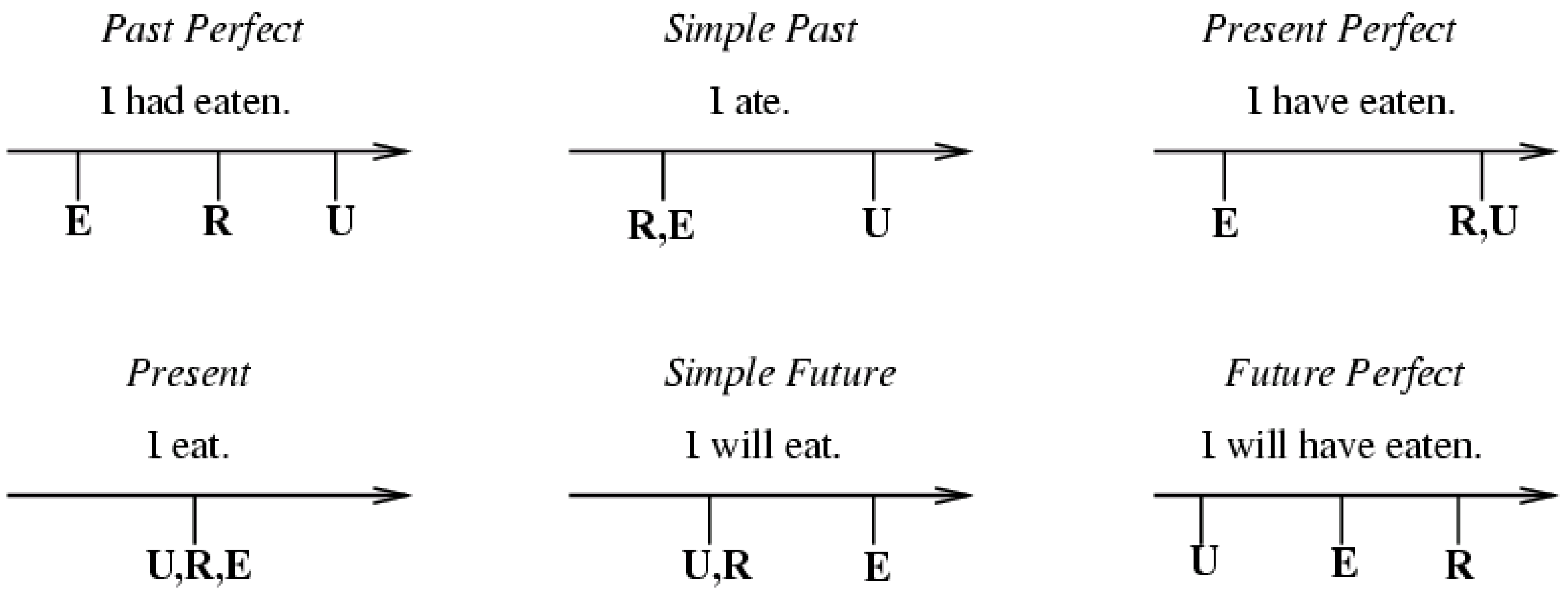
  (14.36) When Mary's flight departed, I ate lunch.

  (14.37) When Mary's flight departed, I had eaten lunch.
- *Departure* event specifies the reference point.

  (14.36)eating event began when the flight departed – reference point *precedes* eating

  (14.37)eating was accomplished prior to the flight's departure – eating *precedes* reference point

Representing Time



Reichenbach's approach to various English tenses:
E – time of event, **R** – reference time, **U** – time of the utterence

Aspect

- **Aspect**
  - Concerning a cluster of related topics, including
    - whether an event has ended or is ongoing,
    - whether it is conceptualized as happening at a point in time or over some interval, and
    - Whether or not any particular state in the world comes about because of it.

Aspect

- Four general classes
  - **Stative:** represent the notion of an event participant having a particular property, or being in a state at a given point in time
  - **Activity:** describe events undertaken by a participant that have no particular end point. Activities are seen as occurring over some span of time
  - **Accomplishment:** describes events that have a natural end point and result in a particular state.
  - **Achievement:** events are thought of as happening in an instant, and are not equated with any particular activity.

Aspect

- Four general classes: Examples
- **Stative:** I know my departure time.
- **Activity:** can be used with *for* temporal adverb

  John is flying. She drove a Mazda. He is living in Calicut.

  I live in Chennai for a month.
- **Accomplishment:** can be used with *in* temporal adverb

  Ramesh booked his flight. She booked a flight in a minute.
- **Achievement:** She found her gate. I reached New York.

Representing Beliefs

- Words and expressions for *world creating activity*
  - Their meaning representations contain logical formulas not intended to be taken as true in the real world, but rather as part of some kind of hypothetical world.
  - For example, *believe*, *want*, *imagine*, and *know*
  - (14.72) I believe that Mary ate British food.
  - Event-oriented approach: two events – believing event, eating event

$$\exists\, u, v\ ISA(u, Believing) \wedge ISA(v, Eating)$$
$$\wedge\ Believer(u, Speaker) \wedge BelievedProp(u, v)$$
$$\wedge\ Eater(v, Mary) \wedge Eaten(v, BritishFood)$$

  - This results in a statement that there actually was an eating of British food by Mary.

Representing Beliefs

- *Believing*(*Speaker*, *Eating*(*Mary, BritishFood*))
- Problem: It is not even valid FOPC.
- Solution: Augment FOPC with *operators* that allow us to make statements about full logic formulas.
  - Operator *Believes* takes two FOPC formulas as it arguments
    - A formula designating a believer, and
    - A formula designating the believed propositions.

$Believes(Speaker, \exists v\ ISA(v, Eating) \land Eater(v, Mary) \land Eaten(v, BritishFood))$

References

- Speech and Language Processing, *Jurafsky and H.Martin*
  [Chapter 14. Representing Meaning]