From NL to FOL

Scott Farrar
CLMA, University of Washington
farrar@u.washington.edu

February 17, 2010
Today's lecture

1. Review

2. From NL to Logic

3. Semantics and the NLTK
What is/are:

- the three main problems associated with computational semantics?
- a logic?
- some logics we've talked about?
- a logical sentence?
- the logical connectives of sentential logic?
- a logical argument?
- a tautology?
What is/are:

- the **three main problems** associated with computational semantics?
What is/are:

- the **three main problems** associated with computational semantics?
- a **logic**?
What is/are:

- the **three main problems** associated with computational semantics?
- a **logic**?
- some logics we’ve talked about?
What is/are:

- the **three main problems** associated with computational semantics?
- a **logic**?
- some logics we’ve talked about?
- a **logical sentence**?
What is/are:

- the **three main problems** associated with computational semantics?
- a **logic**?
- some logics we’ve talked about?
- a **logical sentence**?
- the **logical connectives** of sentential logical?
What is/are:

- the **three main problems** associated with computational semantics?
- a **logic**?
- some logics we’ve talked about?
- a **logical sentence**?
- the **logical connectives** of sentential logical?
- a logical **argument**?
What is/are:
- the **three main problems** associated with computational semantics?
- a **logic**?
- some logics we’ve talked about?
- a **logical sentence**?
- the **logical connectives** of sentential logical?
- a logical **argument**?
- a **tautology**?
What is meaning, really?

An example:

1. English:
   Horatio bakes me a cake
   or
   Horatio bakes you a cake.

2. Sentential Logic:
   $P \lor Q$
   (just a symbolic translation)

Intuitive approach to meaning:
Intuitively we say that (1)'s meaning depends on whether Horatio bakes a cake, and whether it's for you or me. That is, depending on the state of the world. Not very satisfying!
What is meaning, really?

An example:

1. English: *Horatio bakes me a cake or Horatio bakes you a cake.*
What is meaning, really?

An example:

1. English: *Horatio bakes me a cake or Horatio bakes you a cake.*
2. Sentential Logic: $P \lor Q$ (just a symbolic translation)
What is meaning, really?

An example:

1. English: *Horatio bakes me a cake* or *Horatio bakes you a cake*.

2. Sentential Logic: \[ P \lor Q \] (just a symbolic translation)

Intuitive approach to meaning
What is meaning, really?

An example:

1. English: *Horatio bakes me a cake* or *Horatio bakes you a cake.*
2. Sentential Logic: $P \lor Q$ (just a symbolic translation)

Intuitive approach to meaning

- Intuitively we say that (1)’s meaning depends on whether Horatio bakes a cake, and whether it’s for you or me.
What is meaning, really?

An example:

1. English: *Horatio bakes me a cake or Horatio bakes you a cake.*
2. Sentential Logic: $P \lor Q$ (just a symbolic translation)

Intuitive approach to meaning

- Intuitively we say that (1)’s meaning depends on whether Horatio bakes a cake, and whether it’s for you or me.
- That is, depending on the state of the world.
What is meaning, really?

An example:

1. English: *Horatio bakes me a cake* or *Horatio bakes you a cake.*
2. Sentential Logic: $P \lor Q$ (just a symbolic translation)

Intuitive approach to meaning

- Intuitively we say that (1)’s meaning depends on whether Horatio bakes a cake, and whether it’s for you or me.
- That is, depending on the state of the world.
- Not very satisfying!
Slightly more formal account

What about assigning a value to meaning?

Something like the meaning of *Horatio bakes me a cake.* is:
What about assigning a value to meaning?

Something like the meaning of *Horatio bakes me a cake.* is:

- **True**
Slightly more formal account

What about assigning a value to meaning?

Something like the meaning of *Horatio bakes me a cake.* is:

- True
- 0.43
What about assigning a value to meaning?

Something like the meaning of *Horatio bakes me a cake.* is:

- True
- 0.43

Truth values

In fact, we can say that the atomic sentences of our logic $P$, $Q$, ..., $X$ can either be **True** or **False** depending on the state of the world. This is called a **truth functional logic**.
Slightly more formal account

What about assigning a value to meaning?

Something like the meaning of *Horatio bakes me a cake.* is:
- True
- 0.43

Truth values

In fact, we can say that the atomic sentences of our logic $P$, $Q$, ..., $X$ can either be True or False depending on the state of the world. This is called a truth functional logic.

Complex sentences

What about $P \lor Q$? When is it True or False?
Truth table for ‘inclusive or’, $\lor$

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$P \lor Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Truth table for ‘exclusive or’, $\oplus$

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$P \oplus Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Today’s lecture

1. Review

2. From NL to Logic

3. Semantics and the NLTK
Semantic analysis is the mapping of NL utterances onto some logic. In traditional logic classes, the mapping is usually done from logic to NL:
Semantic analysis is the mapping of NL utterances onto some logic. In traditional logic classes, the mapping is usually done from logic to NL:

Example

\[ \exists x \ (\text{dog}(x) \land \text{disappear}(x)) \]

At least one entity is a dog and disappeared.

Some dog disappeared.

A dog disappeared.
Semantic analysis is the mapping of NL utterances onto some logic. In traditional logic classes, the mapping is usually done from logic to NL:

\[ \exists x \ (\text{dog}(x) \land \text{disappear}(x)) \]
At least one entity is a dog and disappeared.
Some dog disappeared.
A dog disappeared.

Our methodology: start with structures in NL and find appropriate logical formulas. This makes the logic work for NL, not the other way around.
Mapping NL to FOL: Nouns

What do **nouns** usually denote?

- physical objects
- abstract objects
- events

- fish, dogs, hat, leg
- value, politics, mathematics, scorn
- destruction, creation, movement
Mapping NL to FOL: Nouns

What do nouns usually denote?

- **physical objects**, ones that exist in time and space

- fish, dogs, hat, leg
Mapping NL to FOL: Nouns

What do *nouns* usually denote?

- **physical objects**, ones that exist in time and space
- **abstract objects**, ones that may persist in time, but not space

- fish, dogs, hat, leg
- value, politics, mathematics, scorn
Mapping NL to FOL: Nouns

What do **nouns** usually denote?

- **physical objects**, ones that exist in time and space
- **abstract objects**, ones that may persist in time, but not space
- **events**, ones that exist in time and space, but not as long as objects

- fish, dogs, hat, leg
- value, politics, mathematics, scorn
- destruction, creation, movement
Mapping NL to FOL: Nouns

Unary predicates, those with a single argument, are often used to represent the semantics of nouns.

fish(FLIPPER), dog(FRITZ), hat(HAT234), value(V1), politics(P1), scorn(S1), destroyingEvent(D1), movementEvent(M1)

In fact the unary predicate is naming the **type** of whatever the argument may be. There can be many, many types wrt the semantics of natural language.
NL semantics and ontology

Definition

Our FOL approach requires a rich **ontology**, or a theory of existence and how the elements of the world relate.
NL semantics and ontology

Definition

Our FOL approach requires a rich **ontology**, or a theory of existence and how the elements of the world relate.

Modeling NL meaning is an exercise in ‘natural language’ metaphysics.
DOLCE upper categories
Mapping NL to FOL: PNs

Constants are used to represent the semantics of proper nouns.

- Queen Elizabeth II, ELIZII
- Barack Obama, BARACK
- John, JOHN432
Mapping NL to FOL: NPs

A modified NP is an NP of the form: $NP \rightarrow X \ NN$, where $X$ can be one of a number of syntactic categories: determiner, quantity, adjective, etc.

- a dog, some dog
- all cats
- 3 fish
- several elephants
- a bunch of rats
Quantificational modifiers of the type \textit{a} or \textit{some} are modeled using the existential quantifier, $\exists$.

- a dog, $\exists x \text{ dog}(x)$
- some person, $\exists x \text{ person}(x)$
And the quantificational modifier of the type *all, every, etc.* can be modeled using the universal $\forall$: 
And the quantificational modifier of the type \textit{all}, \textit{every}, etc. can be modeled using the universal $\forall$:

- all fish, $\forall f \ fish(f)$
And the quantificational modifier of the type \textit{all}, \textit{every}, etc. can be modeled using the universal $\forall$:

- all fish, $\forall f \ fish(f)$
- every man, $\forall m \ man(m)$
And the quantificational modifier of the type **all, every, etc.** can be modeled using the universal \( \forall \):

- all fish, \( \forall f \ fish(f) \)
- every man, \( \forall m \ man(m) \)
- each and every member, \( \forall m \ member(m) \)
Mapping NL to FOL: Quantificational modifiers

In reality quantification in languages is often difficult to describe given the standard logical machinery of FOL: \( \exists \) and \( \forall \).
Mapping NL to FOL: Quantificational modifiers

In reality quantification in languages is often difficult to describe given the standard logical machinery of FOL: $\exists$ and $\forall$.

Semanticists are often forced to come up with new logical quantifiers to discuss natural language quantification.
In reality quantification in languages is often difficult to describe given the standard logical machinery of FOL: $\exists$ and $\forall$.

Semanticists are often forced to come up with new logical quantifiers to discuss natural language quantification.
In reality quantification in languages is often difficult to describe given the standard logical machinery of FOL: $\exists$ and $\forall$.

Semanticists are often forced to come up with new logical quantifiers to discuss natural language quantification.

- Just a few biscuits please, with a lot of gravy!
Mapping NL to FOL: Quantificational modifiers

In reality quantification in languages is often difficult to describe given the standard logical machinery of FOL: ∃ and ∀.

Semanticists are often forced to come up with new logical quantifiers to discuss natural language quantification.

- Just a few biscuits please, with a lot of gravy!
- He’ll take several doses to be cured.
In reality quantification in languages is often difficult to describe given the standard logical machinery of FOL: $\exists$ and $\forall$.

Semanticists are often forced to come up with new logical quantifiers to discuss natural language quantification.

- Just a **few** biscuits please, with a **lot** of gravy!
- He’ll take **several** doses to be cured.
- A **couple** of slices.
Mapping NL to FOL: Quantificational modifiers

In reality quantification in languages is often difficult to describe given the standard logical machinery of FOL: ∃ and ∀.

Semanticists are often forced to come up with new logical quantifiers to discuss natural language quantification.

- Just a few biscuits please, with a lot of gravy!
- He’ll take several doses to be cured.
- A couple of slices.
- Simply scads of weapons of mass destruction.
Mapping NL to FOL: Adjectives

Just as with nouns, adjectives say something about the type of entity being referred to; accordingly, they can be modeled as unary predicates at the semantic level.
Mapping NL to FOL: Adjectives

Just as with nouns, adjectives say something about the type of entity being referred to; accordingly, they can be modeled as unary predicates at the semantic level.

\[ \text{broken leg}, \quad \text{broken}(x) \wedge \text{leg}(x) \]
\[ \text{red rooster}, \quad \text{red}(y) \wedge \text{rooster}(y) \]
\[ \text{jagged white pill}, \quad \text{jagged}(z) \wedge \text{white}(z) \wedge \text{pill}(z) \]

Some adjectives are more problematic and require more elaborate semantic machinery:

\[ \text{small solar system}, \quad \text{large mouse}, \quad \text{grande latte} \]
Mapping NL to FOL: Adjectives

Just as with nouns, adjectives say something about the type of entity being referred to; accordingly, they can be modeled as unary predicates at the semantic level.

- broken leg, $\text{broken}(x) \land \text{leg}(x)$
Mapping NL to FOL: Adjectives

Just as with nouns, adjectives say something about the type of entity being referred to; accordingly, they can be modeled as unary predicates at the semantic level.

- broken leg, $\text{broken}(x) \land \text{leg}(x)$
- red rooster, $\text{red}(y) \land \text{rooster}(y)$
Mapping NL to FOL: Adjectives

Just as with nouns, adjectives say something about the type of entity being referred to; accordingly, they can be modeled as unary predicates at the semantic level.

- broken leg, $\text{broken}(x) \land \text{leg}(x)$
- red rooster, $\text{red}(y) \land \text{rooster}(y)$
- jagged white pill, $\text{jagged}(z) \land \text{white}(z) \land \text{pill}(z)$
Mapping NL to FOL: Adjectives

Just as with nouns, adjectives say something about the type of entity being referred to; accordingly, they can be modeled as unary predicates at the semantic level.

- broken leg, \( \text{broken}(x) \land \text{leg}(x) \)
- red rooster, \( \text{red}(y) \land \text{rooster}(y) \)
- jagged white pill, \( \text{jagged}(z) \land \text{white}(z) \land \text{pill}(z) \)

Some adjectives are more problematic and require more elaborate semantic machinery: small solar system, large mouse, grande latte.
Mapping NL to FOL: Conjunctions

Conjunctions such as **and**, **as well as**, **or**, **either...or...** are mapped to logical connectives.
Mapping NL to FOL: Conjunctions

Conjunctions such as *and, as well as, or, either...or...* are mapped to logical connectives.

- fish and chips, \( fish(f) \land chip(c) \)
Mapping NL to FOL: Conjunctions

Conjunctions such as **and**, **as well as**, **or**, **either...or...** are mapped to logical connectives.

- Fish and chips, \( fish(f) \land chip(c) \)
- Ham as well as eggs, \( ham(h) \land egg(e) \)
Mapping NL to FOL: Conjunctions

Conjunctions such as **and**, **as well as**, **or**, **either...or...** are mapped to logical connectives.

- fish and chips, $fish(f) \land chip(c)$
- ham as well as eggs, $ham(h) \land egg(e)$
- coffee or tea, $coffee(c) \lor tea(t)$
Mapping NL to FOL: Conjunctions

Conjunctions such as **and, as well as, or, either...or...** are mapped to logical connectives.

- fish and chips, $fish(f) \land chip(c)$
- ham as well as eggs, $ham(h) \land egg(e)$
- coffee or tea, $coffee(c) \lor tea(t)$
- either red or green, $red(x) \oplus green(x)$ (XOR)
Mapping NL to FOL: misc

Negative markers are mapped to formulas with the negation symbol.
not pumpkin, $\neg pumpkin(p)$
Mapping NL to FOL: misc

Negative markers are mapped to formulas with the negation symbol.  
not pumpkin, $\neg pumpkin(p)$

Some prepositions are mapped to binary predicates.  
Joe is in Seattle, $in(JOE, SEATTLE)$
**Mapping NL to FOL: misc**

Negative markers are mapped to formulas with the negation symbol.  
not pumpkin, $\neg pum\text{pk}(p)$

Some prepositions are mapped to binary predicates.  
Joe is in Seattle, $in(JOE, SEATTLE)$

Copulas (certain occurrences of *be*) are mapped to equality.  
Fred is the killer, $FRED = KILLER123$
Mapping NL to FOL: Verbs

The main verb is mapped to an $n$-ary predicate in FOL. It indicates a property or relation.
Mapping NL to FOL: Verbs

The main verb is mapped to an $n$-ary predicate in FOL. It indicates a property or relation.

- Intransitives can be represented as unary predicates. $swim(x)$
The main verb is mapped to an $n$-ary predicate in FOL. It indicates a property or relation.

- Intransitives can be represented as unary predicates. $swim(x)$
- Transitives can be represented as binary predicates. $steal(x, y)$
Mapping NL to FOL: Verbs

The main verb is mapped to an \( n \)-ary predicate in FOL. It indicates a property or relation.

- Intransitives can be represented as unary predicates. 
  \( swim(x) \)
- Transitives can be represented as binary predicates. 
  \( steal(x, y) \)
- Ditransitives can be represented as ternary predicates. 
  \( give(x, y, z) \)
Mapping NL to FOL: Verbs

The main verb is mapped to an $n$-ary predicate in FOL. It indicates a property or relation.

- Intransitives can be represented as unary predicates. $swim(x)$
- Transitives can be represented as binary predicates. $steal(x, y)$
- Ditransitives can be represented as ternary predicates. $give(x, y, z)$

But there’s a problem.
Mapping NL to FOL: VPs

- John buttered the toast.
  
  \textit{Butter}(JOHN, TOAST)
Mapping NL to FOL: VPs

- John buttered the toast.
  \( \text{Butter}(\text{JOHN}, \text{TOAST}) \)

- John buttered the toast at midnight.
  \( \text{Butter}(\text{JOHN}, \text{TOAST}, \text{MIDNIGHT}) \)
Mapping NL to FOL: VPs

- John buttered the toast.
  \[\text{Butter}(JOHN, \text{TOAST})\]

- John buttered the toast at midnight.
  \[\text{Butter}(JOHN, \text{TOAST}, \text{MIDNIGHT})\]

- John buttered the toast at midnight with a knife.
  \[\text{Butter}(JOHN, \text{TOAST}, \text{MIDNIGHT}, \text{KNIFE})\]
Mapping NL to FOL: VPs

- John buttered the toast.
  \[\text{Butter}(\text{JOHN}, \text{TOAST})\]

- John buttered the toast at midnight.
  \[\text{Butter}(\text{JOHN}, \text{TOAST}, \text{MIDNIGHT})\]

- John buttered the toast at midnight with a knife.
  \[\text{Butter}(\text{JOHN}, \text{TOAST}, \text{MIDNIGHT}, \text{KNIFE})\]

- John buttered the toast at midnight with a knife before he went to bed.
  \[\text{Butter}(\text{JOHN}, \text{TOAST}, \text{MIDNIGHT}, \text{KNIFE}, ...)]\]
Today’s lecture

1. Review

2. From NL to Logic

3. Semantics and the NLTK
Packages

- `nltk.sem.logic`: A version of first order logic, built on top of the untyped lambda calculus.
- `nltk.sem.logic.Expression` is the base class for all kinds of logical formulas.
- `nltk.sem.LogicParser`: A parser that reads semantic representations and creates logic objects.
- `nltk.sem.util`: Utility functions for batch-processing sentences: parsing and extraction of the semantic representation of the root node of the syntax tree, followed by evaluation of the semantic representation in a first-order model.
- `Prover9`: a supplemental theorem prover callable from the NLTK
FOL in the NLTK

The `nltk.sem.logic` package contains the tools necessary for representing FOL. Boolean operators:

- negation: `~` (the hyphen)
- conjunction: `&`
- disjunction: `|`
- implication: `->`
- equivalence: `<->`
FOL in the NLTK

Equality predicates:

equality =
inequality !=

Quantifiers:

existential exists
universal all
lambda \