Compositional Semantics: Montagovian Semantics and Lambda Calculus

COMP-599

Oct 27, 2015

Announcements

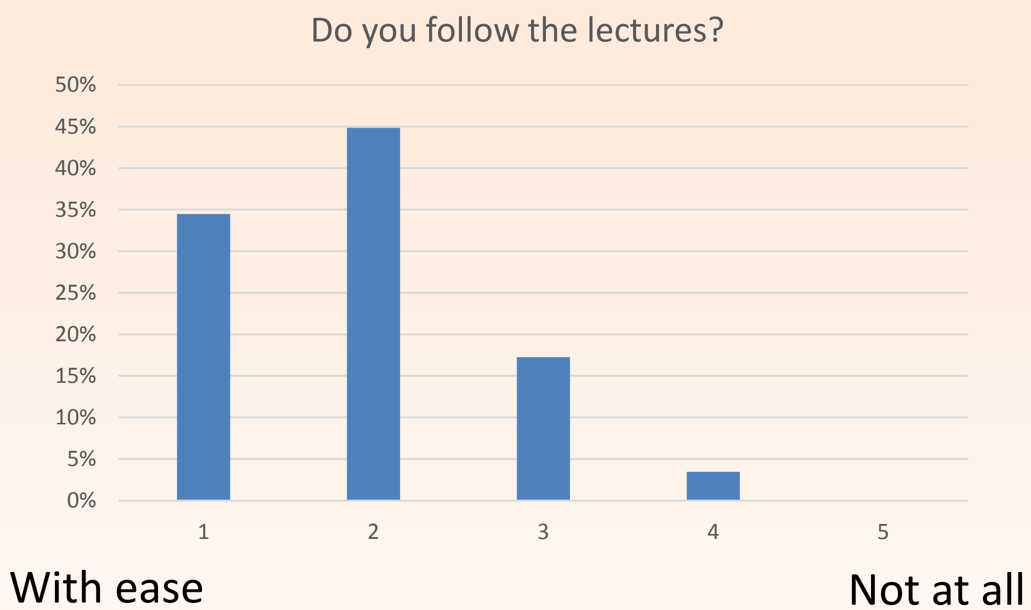
Proposal

- I made several comments
- Please ask me or others for help, especially with data sets!

Office hours

- Today, after class in this room, from 2:30pm-3pm

Class Survey: Pace



Suggestions:

- Ask clarification questions. Don’t be shy!
- Take notes during class

Topics After Midterm

Summarization/NLG	24
Machine Translation	15
Speech	7
Universal Grammar	4
Other responses	9

- Including 2 for cognitive modelling

Result:

- 2 lectures on summarization/NLG
- 2 lectures on MT

Outline

Principle of compositionality

Semantic inference

First-order logic

Lambda calculus

The Principle of Compositionality

Compositionality: The meaning of a phrase depends on the meanings of its parts.

COMP-599 is a fantastically awesome class.

Lexical semantics gives us the meanings of each of the words:

- *COMP-599, is, a, fantastically, awesome, class*

We build up the meaning of the entire sentence through composition.

Idioms – Violation of Compositionality

Idioms are expressions whose meanings cannot be predicted from their parts.

kick the bucket

the last straw

piece of cake

hit the sack

Co-Compositionality

Consider the meaning of *red* when modifying each of the following nouns

- *rose*
- *wine*
- *cheeks*
- *hair*

Is red really combining compositionally with each of these nouns?

- **Co-compositionality** (Pustejovsky, 1995) – the meanings of words depend also on the other words that they are composed with

Goal of Compositional Semantics

Derive a meaning representation of a phrase/sentence from its parts

What is a good meaning representation?

Relates the linguistic expression to the world:

- Asserts a proposition that is either true or false relative to the world

  Pandas are purple and yellow.

- Conveys information about the world

  It will snow tomorrow.

- Is a query about the state of the world

  What is the weather like in Montreal next week?

Semantic Inference

Making *explicit* something that is *implicit* in language (Blackburn and Bos, 2003)

I want to visit the capital of Italy.

The capital of Italy is Rome.

∴ *I want to visit Rome.*

All wugs are blorks.

All blorks are cute.

∴ *All wugs are cute.*

Montagovian Semantics

Montague (1970) started a tradition of using a logical formalism to represent the meaning of a sentence, with a tight connection to syntax.

> *There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed I consider it possible to comprehend the syntax and semantics of both kinds of languages with a single natural and mathematically precise theory.* (Montague 1970c, 222)

Natural language inference then can be seen as applying logical rules of inference.

First-Order Predicate Calculus

Domain of discourse

A set of entities that we care about

e.g., the students in the class, the topics we study, classrooms, courses, etc.

Variables

Typically lower-case

Stands for potential elements of the domain

e.g., x, y, z

First-Order Predicate Calculus

Predicates

Maps elements of the domain to truth values

Can be of different valences

e.g. $inCourse(x, y)$: takes in two elements of the domain, returns true if x is a student in course y, false otherwise

Functions

Maps elements to other elements

Can be of different valences

e.g. $instructorOf(x)$: takes x, returns an element corresponding to x's father

What is a valence 0 function?

First-Order Predicate Calculus

Logical connectives

- All the standard ones

  $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

Quantifiers

- Existential $\exists$
- Universal $\forall$

Example Sentences

The capital of Italy is Rome.

$$capitalOf(ITALY) = ROME$$

All wugs are blorks.

$$\forall x.\, wug(x) \rightarrow blork(x)$$

Interpretating FOL

A particular instance of a FOL consists of:

- Predicate and function names and arity
- A set of sentences in FOL using those predicates and functions

An **interpretation** or **model** of a FOL consists of:

Domain of discourse, D

Mapping for the functions to elements of D

Mapping for the predicates to True or False

Exercise

Come up with a FOL characterization of the following:

- Students who study AND do homework will get an A
- Students who only do one of them get a B
- Students who do neither get a C

List the predicates and functions that are necessary. Make constants for the grades (A, B, C).

Come up with an interpretation of this FOL, where you and your neighbours are the elements in the domain of discourse, such that the above FOL formulas are true.

Building Meaning Representations

Target MR: a logical formula in FOL

Still needed:

- A procedure to map sentences to a FOL formula compositionally

Tool: **Lambda calculus**

Lambda Calculus

Basically a way to describe computation using mathematical functions

- The computation we will be doing is to build up a FOL sentence as the meaning representation of a sentence.

Terms in Lambda calculus can be defined recursively:

- A variable (e.g., x)
- $\lambda x. t$, where t is a lambda term
- ts, where t and s are lambda terms

Functional Abstraction and Application

Function application (or **beta reduction**) of term $(\lambda x.t)s$

- Replace all instances of x in t with the expression s

  e.g., $(\lambda x.x + y)2$ simplifies to $2 + y$

  $(\lambda x.xx)(\lambda x.x) = (\lambda x.x)(\lambda x.x) = (\lambda x.x)$

I define these intuitively here, and gloss over some details, but these definitions can be formalized, in order to be precise:

http://arxiv.org/pdf/1503.09060.pdf

Power of Lambda Calculus

They allow us to store partial computations of the MR, as we are composing the meaning of the sentence constituent by constituent.

Whiskers disdained catnip.

disdained	$\lambda x.\lambda y.disdained(y,x)$
disdained catnip	$\big(\lambda x.\lambda y.disdained(y,x)\big)\ catnip$
	$= \lambda y.disdained(y,catnip)$

Whiskers disdained catnip

$$\big(\lambda y.disdained(y,catnip)\big)Whiskers$$

$$= disdained(Whiskers,catnip)$$

Exercises

What is the result of simplifying the following expressions in lambda calculus through beta reduction?

$(\lambda z.z)(\lambda y.y\ y)(\lambda x.x\ a)$

$(((\lambda x.\lambda y.(x\ y))(\lambda y.y))\ w)$

$(\lambda x.x\ x)\ (\lambda y.y\ x)\ z$

Syntax-Driven Semantic Composition

Augment CFG trees with lambda expressions

- Syntactic composition = function application

Semantic attachments:

$A \rightarrow \alpha_1 \ \ldots \alpha_n$ $\{f(\alpha_j.sem, \ldots, \alpha_k.sem)\}$

syntactic composition semantic attachment

Proper Nouns

Proper nouns are FOL constants

$PN \rightarrow COMP599 \qquad \{COMP599\}$

Actually, we will **type-raise** proper nouns

$PN \rightarrow COMP599 \qquad \{\lambda x. x(COMP599)\}$

- It is now a function rather than an argument.
- We will see why we do this next class.

NP rule:

$NP \rightarrow PN \qquad \{PN.sem\}$

Common Nouns

Common nouns are predicates inside a lambda expression of type $\langle e, t \rangle$

- Takes an entity, tells you whether the entity is a member of that class

$N \rightarrow student \qquad \{\lambda x. Student(x)\}$

Let's talk more about common nouns next class when we also talk about quantifiers.

Intransitive Verbs

We introduce an *event variable e*, and assert that there exists a certain event associated with this verb, with arguments.

$$V \rightarrow rules \qquad \{\lambda x.\exists e.Rules(e) \wedge Ruler(e,x)\}$$

Then, composition is

$$S \rightarrow NP\ VP \qquad \{NP.sem(VP.sem)\}$$

Let's derive the representation of the sentence "*COMP-599 rules*"

Neo-Davidsonian Event Semantics

Notice that we have changed how we represent events

Method 1: multi-place predicate

$Rules(x)$

Method 2: Neo-Davidsonian version with event variable

$\exists e. Rules(e) \wedge Ruler(e, x)$

Reifying the event variable makes things more flexible

- Optional elements such as location and time, passives
- Add information to the event variable about tense, modality

Transitive Verbs

Transitive verbs

$V \rightarrow enjoys$

$\{\lambda w. \lambda z. w(\lambda x. \exists e. Enjoys(e) \wedge Enjoyer(e, z) \wedge Enjoyee(e, x))\}$

$VP \rightarrow V\ NP \qquad \{V.sem(NP.sem)\}$

$S \rightarrow NP\ VP \qquad \{NP.sem(VP.sem)\}$

Exercise: verify that this works with the sentence "*Jackie enjoys COMP-599*"

Next Class

Quantifiers and common nouns

Adjectives, adverbs, and modifiers

Underspecification