Compositional Semantics: Montagovian Semantics and Lambda Calculus

Instructor: Jackie CK Cheung

COMP-550

J&M Ch. 14 – 14.3 (1st); J&M Ch. 17 – 17.4 (2nd); J&M Ch. 19 (3rd old version)

Reminders

Midterm next week!

No class on Wednesday; I’ll have extra office hours in MC 108N during class time.

RA2, PA2 are out

Algorithms From Last Class

Lesk's algorithm

Yarowsky's algorithm

Hearst patterns

Bootstrapping with patterns

Term-Context Matrix

Each row is a vector representation of a word

Context words

$$\begin{array}{l|ccccc} & \textit{the} & \textit{was} & \textit{and} & \textit{British} & \textit{linguist} \\ \textit{Firth} & 5 & 7 & 12 & 6 & 9 \\ \textit{figure} & 276 & 87 & 342 & 56 & 2 \\ \textit{linguist} & 153 & 1 & 42 & 5 & 34 \\ \textit{1950s} & 12 & 32 & 1 & 34 & 0 \\ \textit{English} & 15 & 34 & 9 & 5 & 21 \end{array}$$

Co-occurrence counts

Target words

Cosine Similarity

Compare word vectors A and B by

$$\text{sim}(A, B) = \frac{A \cdot B}{||A|| \, ||B||}$$

This corresponds to the cosine of the angle between the two vectors.

Range of values:

-1 Vectors point in opposite directions

0 Vectors are orthogonal

1 Vectors point in the same direction

If vectors are positive (e.g., they’re count vectors), similarity score is between 0 and 1.

Rescaling the Vectors

Instead of raw counts, people usually use a measure of how much two words are correlated with each other, above chance.

Pointwise mutual information (PMI)

$$\mathrm{pmi}(w_1, w_2) = \log \frac{P(w_1, w_2)}{P(w_1)P(w_2)}$$

- Numerator: probability of both words occurring (i.e., in each other's context)
- Denominator: probability of each word occurring in general

Pointwise Mutual Information Example

the occurs 100,000 times in a corpus with 1,000,000 tokens, of which it co-occurs with *linguistics* 300 times. *linguistics* occurs 2,500 times in total.

$$P(the, linguistics) = 0.0003$$

$$P(the) = 0.1$$

$$P(linguistics) = 0.0025$$

$$\text{pmi}(the, linguistics) = \log \frac{0.0003}{0.00025} = 0.26303 \text{ (base 2)}$$

If ratio is < 1, PMI is negative

- People often discard negative values → positive pointwise mutual information (PPMI)

Truncated SVD

Idea: throw out some of the singular values in Σ

Latent semantic analysis

- Apply SVD to compress the term-context matrix while minimizing reconstruction loss
- Removes noise and prevents overfitting of model

$$\underset{|V| \text{ x } c}{X_k} \cong \underset{|V| \text{ x } k}{W_k} \times \underset{k \text{ x } k}{\Sigma_k} \times \underset{k \text{ x } c}{C_k^T} \qquad , k < m$$

- Use rows of W_k as new word representations

Views of Truncated SVD

It can be shown that for any matrix B of at most rank k,

$$\|X - X_k\|_2 \leq \|X - B\|_2$$

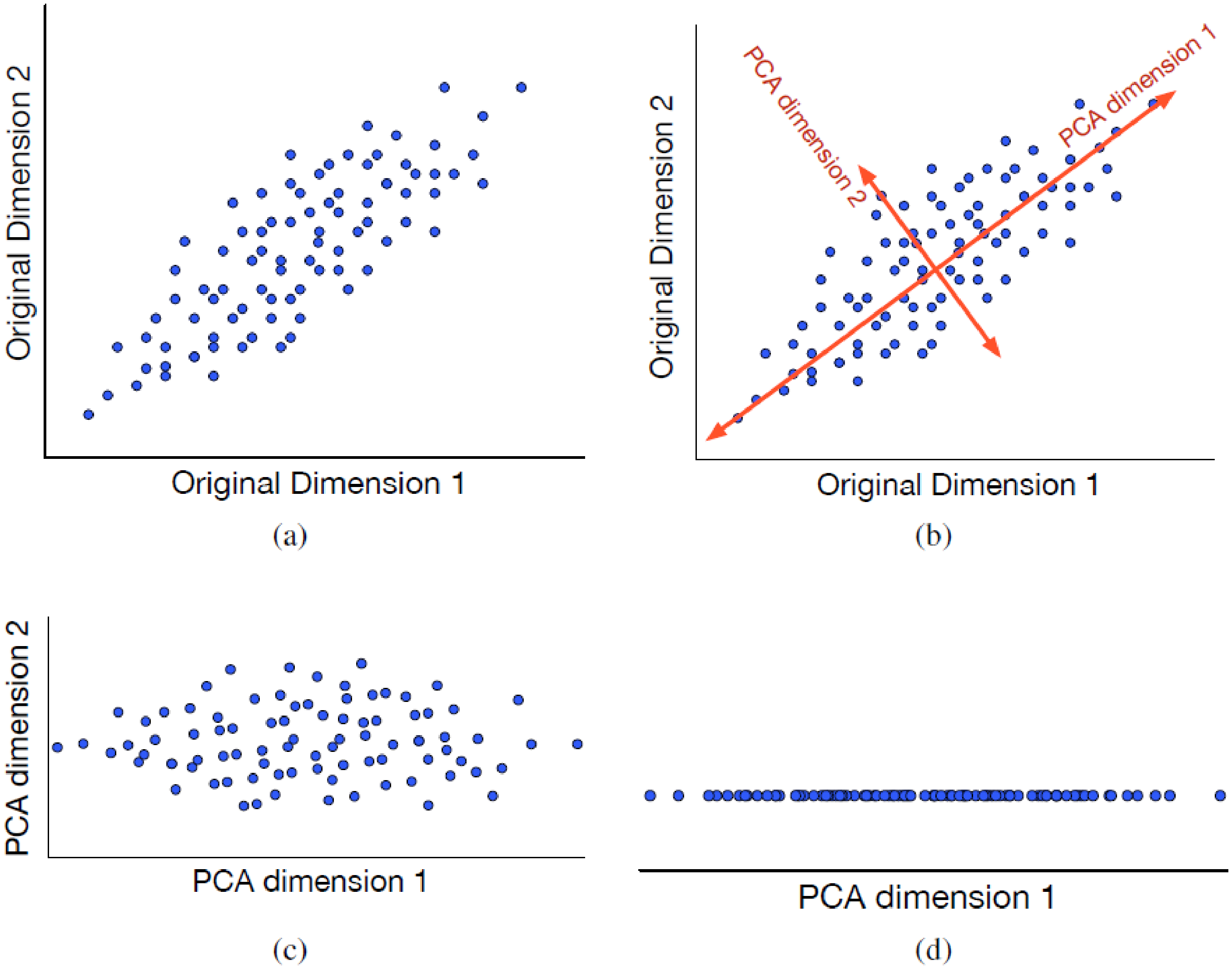
- i.e., X_k is the best possible approximation among any matrix of this rank, according to squared error.

Truncated SVD also corresponds to

1. finding the principal components of the data, then
2. projecting down to the lower-dimensional subspace spanned by these principal components

(**Principal component analysis**)

Principal Components Graphically



J&M 3rd ed.

Word Embeddings

Neural network models – train vector space representation of word to predict words in context

- e.g., word2vec (Mikolov et al., 2013)
- These vector representations of words are called **word embeddings**

- We have a vector of parameters for each word j as a target word v_j, and another vector of parameters for each word as a context word c_j.
- Learn all the v_js and c_js using some auxiliary task

word2vec (Mikolov et al., 2013)

Learn vector representations of words

Actually two models:

- Continuous bag of words (CBOW) – use context words to predict a target word
- Skip-gram – use target word to predict context words

In both cases, the representation that is associated with the target word is the embedding that is learned.

word2vec Architectures

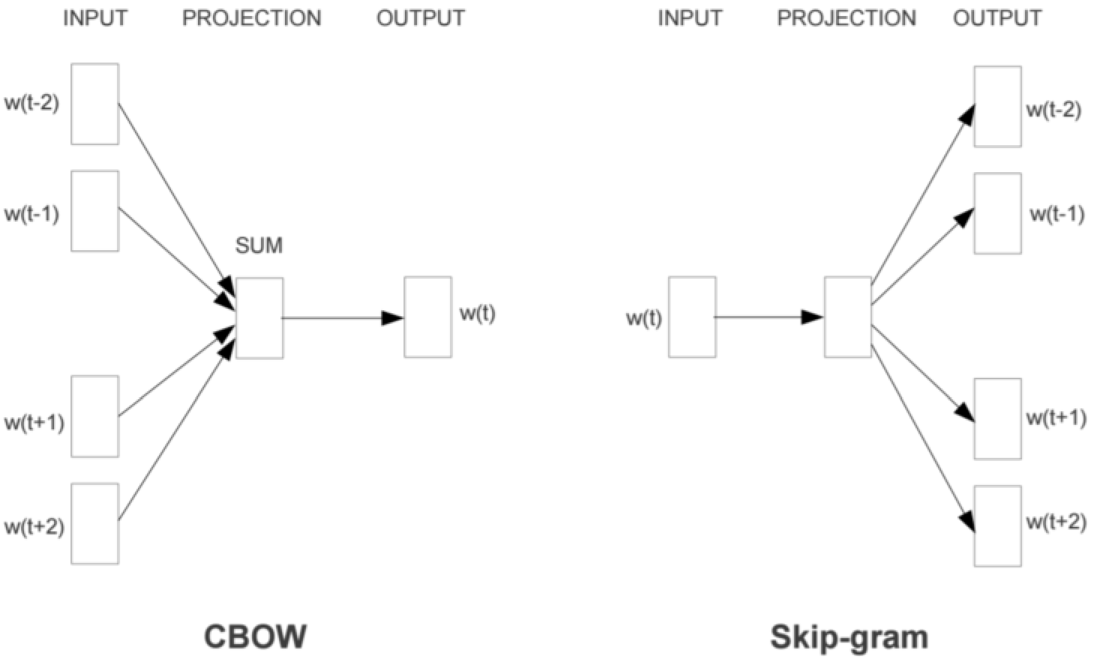


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

(Mikolov et al., 2013)

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-550 is a fantastically awesome class.

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

- *COMP-550, is, a, fantastically, awesome, class*

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

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?

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

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 1970, 222)

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

- So, what is inference?

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.*

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 instructor

What is a valence 0 function?

First-Order Predicate Calculus

Logical connectives

- All the standard ones

  ¬ , ∧ , ∨ , → , ↔

Quantifiers

- Existential ∃
- Universal ∀

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 two of your friends 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 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 $2 + y$
$(\lambda x.xx)(\lambda x.x) = (\lambda x.x)(\lambda x.x) = (\lambda x.x)$

Function application is **left-associative**:

$$abcd = ((ab)c)d$$

I define this notion intuitively here, and gloss over some details, but these definitions can (should) 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	$\left(\lambda x.\lambda y.disdained(y,x)\right)\ catnip$
	$= \lambda y.disdained(y,catnip)$

Whiskers disdained catnip

$$\left(\lambda y.disdained(y,catnip)\right)Whiskers$$

$$= disdained(Whiskers,catnip)$$

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:

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

syntactic composition semantic attachment

Proper Nouns

Proper nouns are FOL constants

$$PN \rightarrow COMP550 \qquad \{COMP550\}$$

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

$$PN \rightarrow COMP550 \qquad \{\lambda x.x(COMP550)\}$$

- 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

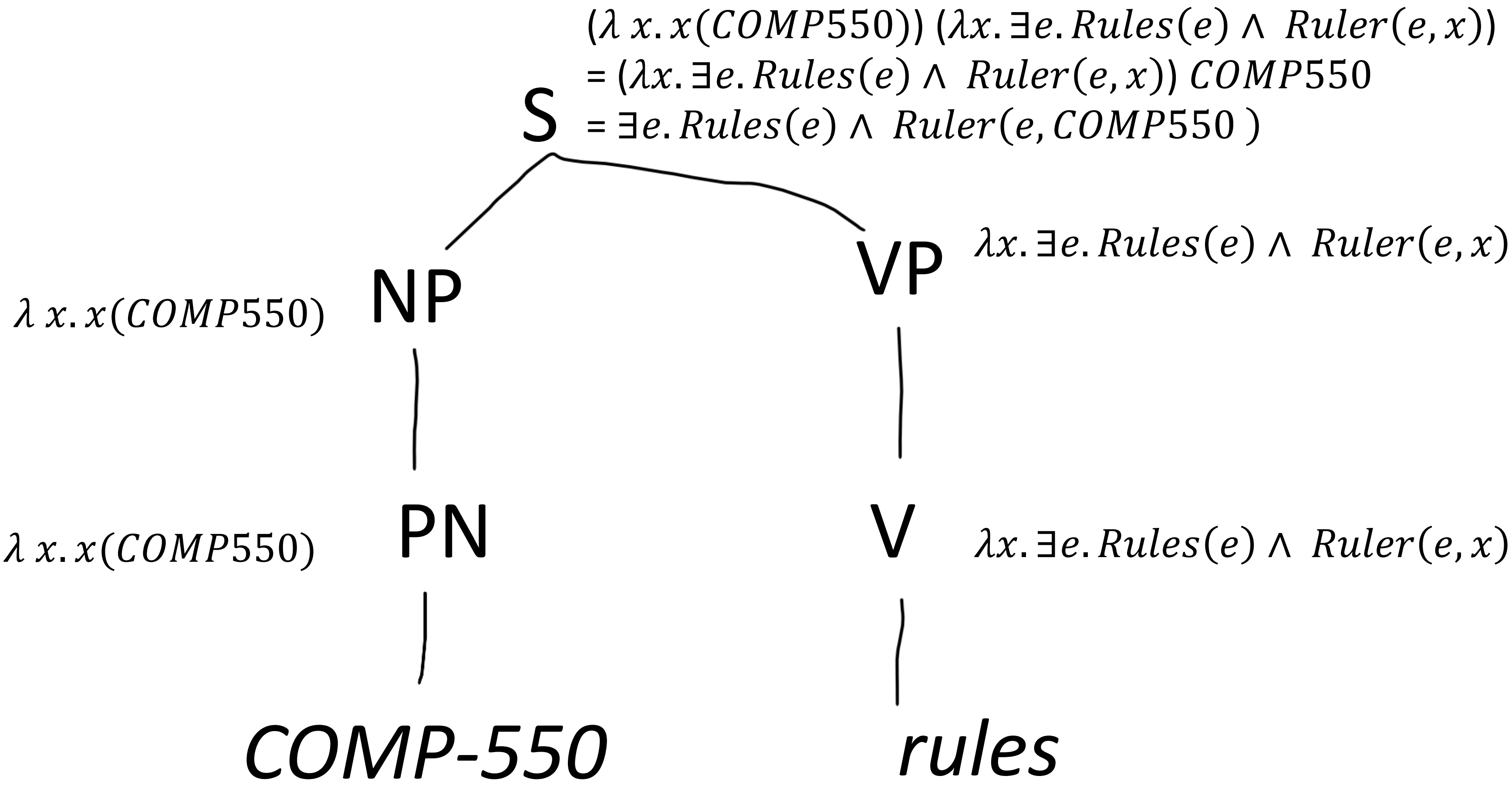
We introduce an *event variable* e, and assert that there exists a certain event associated with this verb, with arguments.

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

Then, composition is

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

Let's derive the representation of the sentence "*COMP-550 rules*"



S

$(\lambda\, x.\, x(COMP550))\ (\lambda x.\exists e.\, Rules(e) \wedge\ Ruler(e,x))$

$= (\lambda x.\exists e.\, Rules(e) \wedge\ Ruler(e,x))\ COMP550$

$= \exists e.\, Rules(e) \wedge\ Ruler(e, COMP550\ )$

NP $\lambda\, x.\, x(COMP550)$

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

PN $\lambda\, x.\, x(COMP550)$

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

COMP-550

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-550*"

Next Class

Quantifiers and common nouns

Adjectives, adverbs, and modifiers

Underspecification