

Text Processing and Corpus Linguistics

UC Davis LIN 127
Spring 2019

Kenji Sagae

Welcome to LIN 127

- Instructor: Kenji Sagae
 - Office hours: Tuesdays 1:30 PM to 2:30 PM
(or by appointment)
Kerr 268

Text Processing and Corpus Linguistics

- What is text processing?
 - What is natural language processing?
 - What is natural language?
 - What is computational linguistics?
 - How is NLP related to Linguistics?
 - What is a corpus?
 - What is corpus linguistics?
-
- Do I really need to know programming and math?!
 - Do I really need to know linguistics?
 - Different students have different strengths and gaps

Assignments and Grading (subject to change)

- Quizzes: 20%
 - Beginning/end of class, 10 minutes
 - No make-up, drop two lowest scores
- Assignments: 40%
 - Painful for some
- Project: 20%
 - Meant to be interesting
- Final: 20%
 - Meant to be straightforward

- Lecture slides will be available on Canvas
- Lecture slides **will not make sense** without the lecture
 - Do not plan to rely on slides to understand the material. Ask questions in class!
- Do not multitask in class
 - Be respectful
 - Don't waste your own time

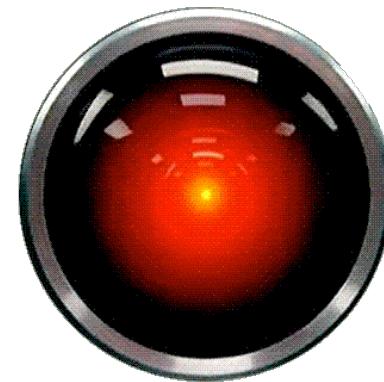
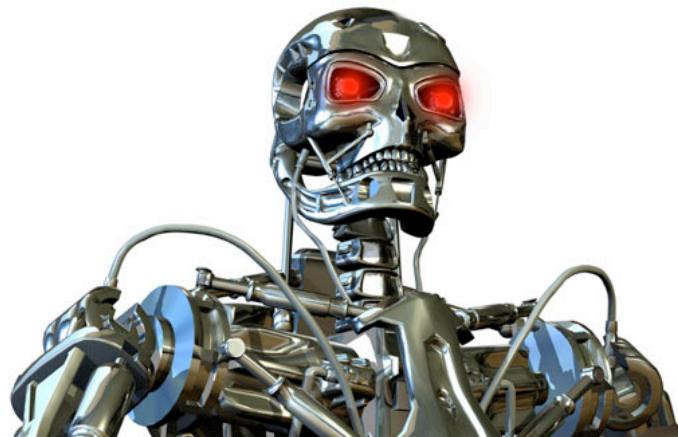
NLP, Word Count, and a First Look at Programming

UC Davis LIN 127
Spring 2019

Kenji Sagae

Natural Language Processing

- What can machines do with natural language?
- How can machines deal with natural language?
- How is NLP related to AI?



Natural Language and AI

- Should an intelligent agent be able to communicate using natural language?
 - Turing test



Language Understanding is Hard

At last, a computer that understands you like your mother.

Example from Lillian Lee's

<http://www.cs.cornell.edu/home/llee/papers/cstb/cmplg.html>

Word Counting is Easy

NLP and Machine Learning

- Classical NLP relied on introspection
 - Rules crafted manually
 - Sometimes (loosely) based on Linguistic theory
 - Very interesting toy examples, scaled poorly
- Many current NLP approaches are data-driven
 - Robust, deals with real-world speech and text
 - Requires training data
 - Learn models from data

Corpus Linguistics

- Study of language through naturally occurring samples
 - 10k words, 100k words... 10B words, even larger
- Quantitative and empirical approach or methodology
 - Not a theory
 - Can describe features and test hypotheses
- Typically involving statistical analysis
 - Counting
 - Typically done by machine
- Many kinds of analysis are possible

A few examples of corpora

- WSJ
 - Text from the Wall Street Journal
- CHILDES
 - Transcripts of dialogues with children
 - Various languages, ages, etc.
- International Corpus of Learner English
 - Writing samples from ESL students
- SCOTUS
 - Supreme court transcripts
- Project Gutenberg
 - Literary works
- Wikipedia, the web, twitter, etc.
- Endless possibilities

LIN 127 Text Processing and Corpus Linguistics: Brief Overview

- You will acquire quantitative and computational skills for addressing language questions using corpora
- NLP & Python programming
- What is a word? (Tokenization)
- How to count words. (Statistics)
- Collection, annotation and analysis of corpora
- Text classification
- Lexical resources, word representations
- Part-of-speech analysis
- Language modeling
- Information extraction

Brief Introduction to Programming (and more)

Computer Program

- A list of instructions to be executed by a computer following a specific order
- For now, let's focus on a simple type of program:

Input → Program → Output

The dog saw the cat → word_count → 5

But how do we know what instructions to use?

Algorithm

- A precisely specified procedure for solving a problem or performing a task step-by-step
- Example: How do we sort a list of integers?
4, 8, 3, 5, 6, 1
- Once we know the algorithm we want to program, we must express it in a programming language
 - Low-level: instructions are close to the level of the basic functionality of the machine
 - High-level: more abstraction, instructions are closer to what people want to express

A Brief introduction to Programming in Python

UC Davis LIN 127
Spring 2019

Kenji Sagae

Computer Program

- A list of instructions to be executed by a computer following a specific order
- For now, let's focus on a simple type of program:

Input → Program → Output

The dog saw the cat → word_count → 5

But how do we know what instructions to use?

Algorithm

- A precisely specified procedure for solving a problem or performing a task step-by-step
- Example: How do we sort a list of integers?
4, 8, 3, 5, 6, 1
- Once we know the algorithm we want to program, we must express it in a programming language
 - Low-level: instructions are close to the level of the basic functionality of the machine
 - High-level: more abstraction, instructions are closer to what people want to express

Operating System

- Manages the computer's resources (e.g. memory, storage, processing time) and execution of programs
- Windows, Mac, Linux
- Shell: user interface to the OS
 - E.g. list files and directories, request program execution, create files, delete files
 - GUI vs command line
- Unix shell
 - 1970's technology!
 - Very useful for simple text analysis with GNU text utilities
 - Just type the name of a program to run it

Input and Output in the Unix Shell

- Standard input
 - Data going into a program, typically from a keyboard
 - But we can use a file
 - Or even the output of another program
- Standard output
 - What the program prints on the screen
 - Can be redirected to a file
 - Can be used as the input to another program

A few simple programs

- Some simple text utilities (Unix shell)
 - cat: prints a file on standard output
 - less: shows text page by page
 - wc: word count
 - sort: sort lines numerically or alphabetically
 - uniq: remove consecutive duplicate lines
 - grep: look for a specific word or pattern in a file
 - tr: replace all instances of a character with another
 - sed: replace, edit a text stream

(and much more!)

Demo

(explanation available on canvas)

Python for text analysis

- Install Python 3.7 and NLTK
- Interactive shell
 - Type and see the results immediately
- Programs
 - Write a list of instructions to be executed
- NLTK: the natural language toolkit for Python
 - Includes ready-made functionality for many text analysis tasks

Types of Values

- Integer
 - 1, 2, -5, 0
- Float
 - 23.452, -0.734
- String
 - ‘hello world!’
 - “hello world!”

Variables

- `x = 5`
- `number_of_students = 49`
- `average_score = 87.245`
- `instructor = "Kenji Sagae"`
- `class_name = 'LIN127'`

- Operators
 - $y = x + 2$

- `print(x)`
- `print(y)`
- `print(class_name)`

Common types of instructions

- Input, output
 - Get data, display data
- Mathematical operations
- Control flow
 - Conditionals
 - Execute these instructions only if a condition holds
 - Loop
 - Keep executing these instructions
 - For several times
 - Or while some condition holds
 - Or forever

Input and Output

- Output can be written to the terminal (screen)
 - `print("Hello world")`
- Output can be written to a file directly (more later)
- Input can be from the keyboard (or from standard input)
 - `name = input("What's your name?")`
- Input can come from a file (more later)
- Input/Output redirection (from the command prompt/shell)
 - `myprogram < input.txt`
 - `myprogram > output.txt`
 - `myprogram < input.txt > output.txt`
 - (.txt extension is not required and could be anything!)

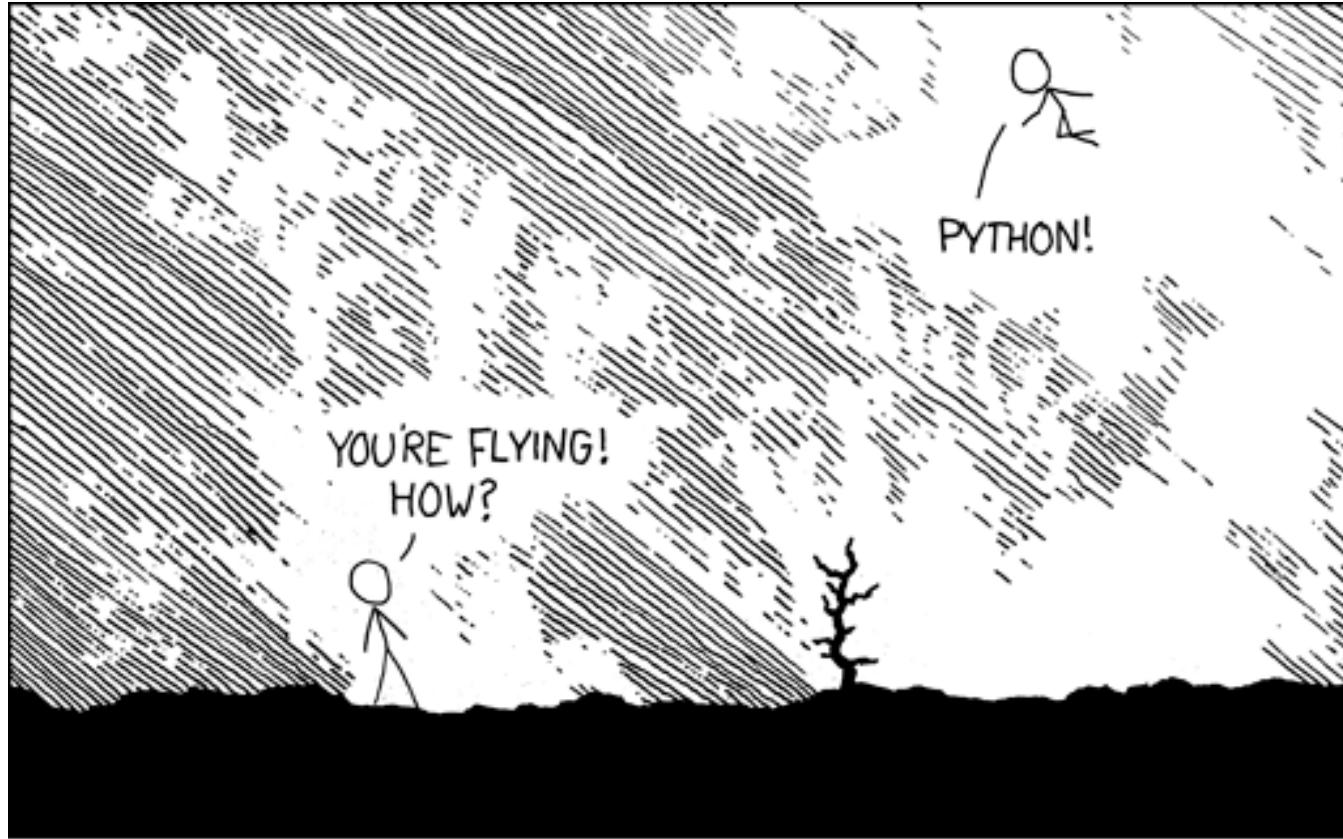
Operators

- Arithmetic operators
 - +, -, /, *, ** (exponentiation), % (modulus), //
- Assignment
 - = (e.g. my_value = 5)
 - +=, etc
- Comparison
 - ==, !=, <, >, <=, >=
- (more operators later)

Python Lists, Dictionaries, NLTK

UC Davis LIN 127
Spring 2019

Kenji Sagae



I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!

HELLO WORLD IS JUST

`print "Hello, world!"`

I DUNNO... DYNAMIC TYPING? WHITESPACE?

COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!

BUT HOW ARE YOU FLYING?

I JUST TYPED

`import antigravity`

THAT'S IT?

... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.

BUT I THINK THIS IS THE PYTHON.

Types of Values

- Integer
 - 1, 2, -5, 0
- Float
 - 23.452, -0.734
- String
 - ‘hello world!’
 - “hello world!”

Variables

- `x = 5`
- `number_of_students = 49`
- `average_score = 87.245`
- `instructor = "Kenji Sagae"`
- `class_name = 'LIN127'`

- Operators
 - `y = x + 2`

- `print(x)`
- `print(y)`
- `print(class_name)`

Common types of instructions

- Input, output
 - Get data, display data
- Mathematical operations
- Control flow
 - Conditionals
 - Execute these instructions only if a condition holds
 - Loop
 - Keep executing these instructions
 - For several times
 - Or while some condition holds
 - Or forever

Input and Output

- Output can be written to the terminal (screen)
 - `print("Hello world")`
- Output can be written to a file directly (more later)
- Input can be from the keyboard (or from standard input)
 - `name = input("What's your name?")`
- Input can come from a file (more later)
- Input/Output redirection (from the command prompt/shell)
 - `myprogram < input.txt`
 - `myprogram > output.txt`
 - `myprogram < input.txt > output.txt`
 - (.txt extension is not required and could be anything!)

Operators

- Arithmetic operators
 - +, -, /, *, ** (exponentiation), % (modulus), //
- Assignment
 - = (e.g. my_value = 5)
 - +=, etc
- Comparison
 - ==, !=, <, >, <=, >=
- (more operators later)

Control Flow

- Programs can be executed from top to bottom
 - Execute each instruction once
- But we can also choose when to skip and repeat
 - Very powerful
 - `print('hello')`
 - if it's morning, `print('good morning')`
 - otherwise, if it's afternoon, `print('good afternoon')`
 - otherwise, `print('good night')`
- (see guessing example from A Byte of Python)

Control Flow

- if, elif, else: test a condition, execute instructions or alternative instructions
- while: keep repeating as long as a condition holds
- for: iterate over a sequence
- Code blocks: python defines the blocks of instructions that are affected by if, while, for, etc. using **indentation**
 - Very important to get the indentation right

Lists

- In Python, list is a data structure that holds an ordered collection of items
 - `my_list = [4, 7, 2, 3, 4, 3, 0, 1]`
 - `fruit = ['orange', 'apple', 'banana']`
 - `fruit[1]`
 - `len(fruit)`
- Useful *methods* for lists:
 - `append`: add more items to the end of the list
 - `sort`: put list in order
 - More:
 - <https://docs.python.org/3/tutorial/datastructures.html>

Text in Lists

- There are many ways to represent a text in a program
 - “The students aced the final.”
 - [“The”, “students”, “aced”, “the”, “final.”]
 - “The students aced the final.”.split()
- Using lists, we can separate each word...
 - [“The”, “students”, “walked.”, “They”, “also”, “ran.”]
- Each sentence...
 - [[“The”, “students”, “walked.”], [“They”, “also”, “ran.”]]
- And so on.
 - Lists inside lists

Lists and Loops

- We can use for loops to iterate through lists
 - `my_text = ["The", "students", "aced", "the", "final."]`
 - `for word in my_text:`
 `print(word)`
- We can loop keeping an index
 - `for index, word in enumerate(my_text):`
 `print(index, word)`

Dictionaries

- Python dictionaries: data indexed by *keys*
 - The keys can be strings!
- Recall lists
 - `my_fruit_list = ['apple', 'lemon', 'banana']`
 - `my_fruit_list[2]`
 - Index is the position in the ordered list
- Indexing with strings (dictionary)
 - `my_fruit_dict = {'apple': 'red', 'lemon': 'yellow', 'banana': 'yellow'}`
 - `my_fruit_dict['apple']`

NLTK

- A toolkit for working with natural language in Python
- Example:

```
import nltk  
raw_text = "The students aced the final."  
tokenized_text = nltk.word_tokenize(raw_text)  
for word in tokenized_text:  
    print(word)
```

Homework 1

- Use Python and NLTK to find:
 - The number of words in President Washington's inaugural address
 - The 20 most frequent words and their counts
 - Same for President Obama's and President Trump's inaugural addresses
- Details are on canvas

Types, Tokens, Text Classification

UC Davis LIN 127
Spring 2019

Kenji Sagae

Functions

- A reusable piece of code
 - Possibly with parameters
 - Returns a value

```
def say_hello_twice():
    print('hello')
    print('hello')
```

```
say_hello_twice()
```

```
def say_hello_name(name):
    print('hello, ' + name)
```

```
say_hello_name('Bob')
```

```
def add_two_numbers(n1, n2):
    result = n1 + n2
    return result
```

```
x = add_two_numbers(4, 5)
print(x)
```

Tokens

- Given a text, we can chop it up into pieces called tokens
 - Are tokens words?
- What is a word?
- How many words are in a piece of text?
- What is tokenization?

The exam was easy

The exam was not easy.

The exam wasn't easy.

Her answer was correct.

The student's answer was correct.

Tokens

- Each individual piece (a contiguous sequence of non-whitespace characters, typically) is a token
 - We can have many identical tokens
- Token is often used to refer to word instances in text

The bird saw the bird.

Types

- Type: collection of all tokens containing exactly the same character sequence

the bird saw the bird

Types:

bird the saw

- <http://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html>

Text Classification

Text Classification

- Learn a function (from training data) that maps input into (discrete) classes
 - Input is represented as *features*
 - E.g. spam detection
 - Classes: Spam vs. Not-Spam (binary classification)
 - Features: “bag-of-words” (all words in the email message, without accounting for order)
 - Training data: collection of email messages marked as spam or not-spam
- Where does training data come from?
 - Expert annotation, crowdsourcing, found data, records from the past, etc.

Why classify text?

- Spam detection
 - Binary classification: spam vs. ham
- Sentiment analysis
 - Binary: positive vs. negative
 - Multiclass: positive, neutral, negative
 - (what else?)
- Assigning topic categories
 - Multiclass: sports, politics, entertainment, etc.
- Authorship ID
- Story identification
- Political inclination
- ...

Classification

- For each training example we have:
A feature vector and a target label

$$x_1, x_2, x_3, \dots, x_n \rightarrow y$$

e.g. x_1 is age, x_2 is gender, x_3 is education level, ...
and y is favorite type of movie (action,
comedy, romance, horror)

(Netflix would have lots of training data, and would be able
to predict preferences of future customers)

Classification

- Given a dataset

$$x_{11}, x_{12}, x_{13}, \dots, x_{1N} \rightarrow y_1$$

$$x_{21}, x_{22}, x_{23}, \dots, x_{2N} \rightarrow y_2$$

...

$$x_{M1}, x_{M2}, x_{M3}, \dots, x_{MN} \rightarrow y_M$$

- We want to estimate $f(x_m) = y_m$

Text Classification

UC Davis LIN 127
Spring 2019

Kenji Sagae

Why classify text?

- Spam detection
 - Binary classification: spam vs. ham
- Sentiment analysis
 - Binary: positive vs. negative
 - Multiclass: positive, neutral, negative
 - (what else?)
- Assigning topic categories
 - Multiclass: sports, politics, entertainment, etc.
- Authorship ID
- Story identification
- Political inclination
- ...

Classification

- For each training example we have:
A feature vector and a target label

$$x_1, x_2, x_3, \dots, x_n \rightarrow y$$

e.g. x_1 is age, x_2 is gender, x_3 is education level, ...
and y is favorite type of movie (action,
comedy, romance, horror)

(Netflix would have lots of training data, and would be able
to predict preferences of future customers)

Classification

- Given a dataset

$$x_{11}, x_{12}, x_{13}, \dots, x_{1N} \rightarrow y_1$$

$$x_{21}, x_{22}, x_{23}, \dots, x_{2N} \rightarrow y_2$$

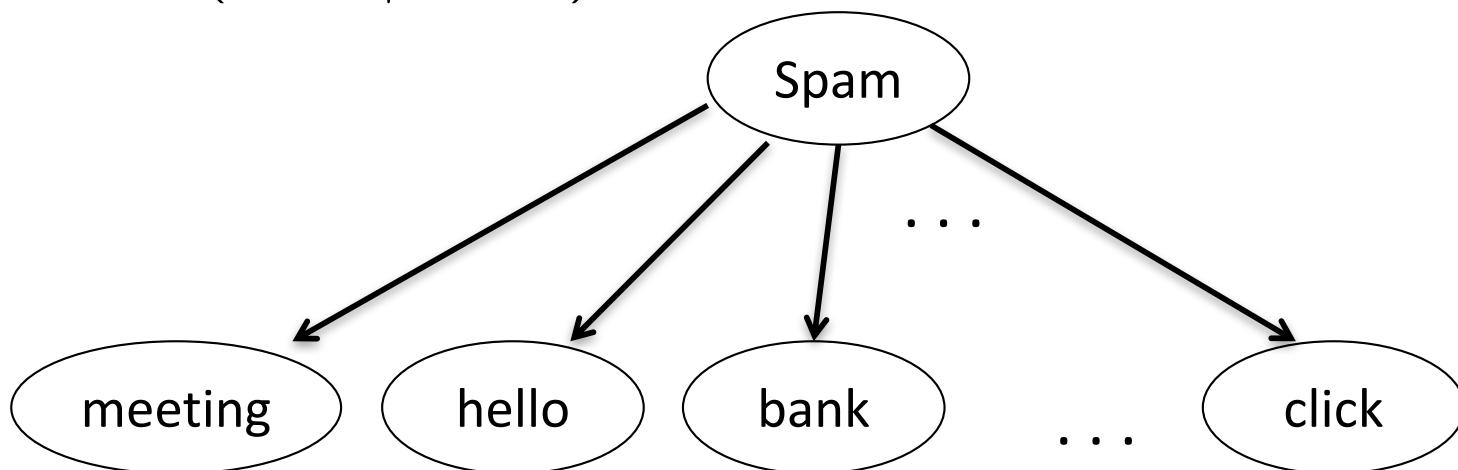
...

$$x_{M1}, x_{M2}, x_{M3}, \dots, x_{MN} \rightarrow y_M$$

- We want to estimate $f(x_m) = y_m$

Naive Bayes Classification for Natural Language

- Naive Bayes classification for **text categorization**
 - A very common baseline, can do surprisingly well
 - Classify news story as sports, business, politics
 - Classify email messages as spam or not-spam
 - Features: bag-of-words
 - All words (with counts, but without order)
 - $P(\text{spam}|\text{words})$



Naive Bayes

- Suppose I want to know if a news article is about sports, politics or entertainment
 - Classes: sports, politics, entertainment
- Probability that a document d belongs to class c
- Probability of class c given document d

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- Compute for every class

Naive Bayes

- Given a document d , what class does it belong to?
 - Find the most likely class c_{pred}

$$\begin{aligned}c_{\text{pred}} &= \arg \max_c P(c \mid d) \\&= \arg \max_c \frac{P(c)P(d \mid c)}{P(d)} \\&= \arg \max_c P(c)P(d \mid c)\end{aligned}$$

$$\begin{aligned}c_{pred} &= \arg \max P(c | d) \\&= \arg \max_c P(c)P(d | c)\end{aligned}$$

- How do we estimate $P(c)$?
- How do we estimate $P(d | c)$?
 - Naive Bayes assumption: words are independent
 - If document d is L words long

$$P(d | c) = P(w_1 | c)P(w_2 | c) \dots P(w_L | c)$$

Spam Filtering with Naive Bayes Classification

- Users create labeled data for free by tagging their own email, so training data is abundant!

SPAM click for pharmacy

OK free time today

SPAM online pharmacy link

OK no free time

OK free good pharmacy

SPAM pharmacy free link

OK for time today

OK time is money

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size:

click	for
pharmacy	free
time	today
online	link
no	good
is	money

12

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12
 $P(\text{spam}) =$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	Maximum Likelihood
OK	no free time	estimate
OK	free good pharmacy	
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12

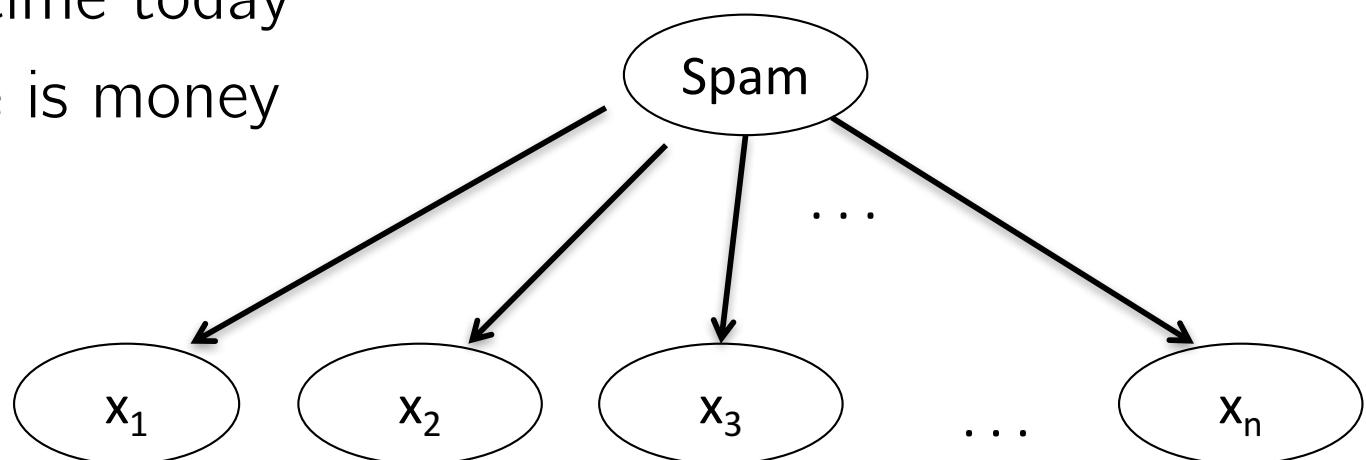
$$P(\text{spam}) = 3/8$$

$$P(\neg \text{spam}) = 5/8$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \mid \text{spam}) =$
OK	free good pharmacy	
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) =$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	



SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) =$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"pharmacy for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(\text{Msg} | \text{spam})}{P(\text{spam}) P(\text{Msg} | \text{spam}) + P(\neg\text{spam}) P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"pharmacy for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(\text{Msg} | \text{spam})}{P(\text{spam}) P(\text{Msg} | \text{spam}) + P(\neg\text{spam}) P(\text{Msg} | \neg\text{spam})}$$

$$P(\text{Msg} | \text{spam}) = P(w_1 | \text{spam}) P(w_2 | \text{spam}) P(w_3 | \text{spam})$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"pharmacy for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{3/8 \ 1/3 \ 1/9 \ 1/3}{P(\text{spam})P(\text{Msg} | \text{spam}) + P(\neg\text{spam})P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"pharmacy for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{1/216 + P(\neg\text{spam})P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for **pharmacy**”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{1/216 + 5/8 \cdot 1/15 \cdot 1/15 \cdot 1/15}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{1/216 + 1/5400}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = 25/26$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"pharmacy for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = 25/26 = 96.15\%$$

What happens if $\text{Msg} = \text{"time for pharmacy"}$?

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"time for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(\text{Msg} | \text{spam})}{P(\text{spam}) P(\text{Msg} | \text{spam}) + P(\neg\text{spam}) P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"time for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(\text{Msg} | \text{spam})}{P(\text{spam})P(\text{Msg} | \text{spam}) + P(\neg\text{spam})P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “time for **pharmacy**”

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam})}{P(\text{spam})P(\text{Msg} | \text{spam}) + P(\neg\text{spam})P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “time for pharmacy”

$$P(\text{spam} | \text{Msg}) = 0$$

Is this classification good?

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"time for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = 0$$

We need smoothing!

e.g. add-one smoothing

Add-one smoothing

- Add one to the count of every observable event
- Suppose X has k possible outcomes, with counts n_1, n_2, \dots, n_k , which sum to N .
- Without smoothing: $P(X = i) = \frac{n_i}{N}$
 - Maximum likelihood estimate
- With add-one smoothing: $P(X = i) = \frac{n_i + 1}{N + k}$
- Not great, but very simple

- Computing $P(c)$
 - Maximum likelihood estimate
 - Number of documents of class c divided by total number of documents
- Computing $P(w | c)$
 - Number of times word w appears in documents in class c , divided by the total number of words in documents in class c
 - With add-one smoothing

$$P(w | c) = \frac{n + 1}{N + k}$$

where n is the number of times w appears in c , N is the total number of words in c , and k is the size of the vocabulary

Naive Bayes Classification, Evaluation

UC Davis LIN 127
Spring 2019

Kenji Sagae

Why classify text?

- Spam detection
 - Binary classification: spam vs. ham
- Sentiment analysis
 - Binary: positive vs. negative
 - Multiclass: positive, neutral, negative
 - (what else?)
- Assigning topic categories
 - Multiclass: sports, politics, entertainment, etc.
- Authorship ID
- Story identification
- Political inclination
- ...

Classification

- For each training example we have:
A feature vector and a target label

$$x_1, x_2, x_3, \dots, x_n \rightarrow y$$

e.g. x_1 is age, x_2 is gender, x_3 is education level, ...
and y is favorite type of movie (action,
comedy, romance, horror)

(Netflix would have lots of training data, and would be able
to predict preferences of future customers)

Classification

- Given a dataset

$$x_{11}, x_{12}, x_{13}, \dots, x_{1N} \rightarrow y_1$$

$$x_{21}, x_{22}, x_{23}, \dots, x_{2N} \rightarrow y_2$$

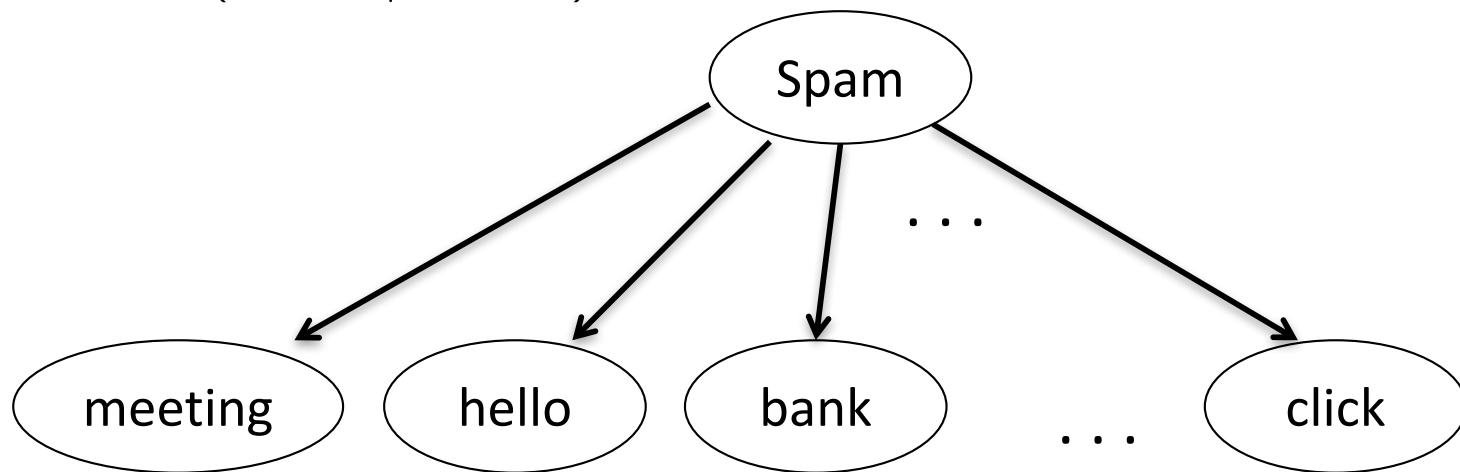
...

$$x_{M1}, x_{M2}, x_{M3}, \dots, x_{MN} \rightarrow y_M$$

- We want to estimate $f(x_m) = y_m$

Naive Bayes Classification for Natural Language

- Naive Bayes classification for **text categorization**
 - A very common baseline, can do surprisingly well
 - Classify news story as sports, business, politics
 - Classify email messages as spam or not-spam
 - Features: bag-of-words
 - All words (with counts, but without order)
 - $P(\text{spam}|\text{words})$



Naive Bayes

- Suppose I want to know if a news article is about sports, politics or entertainment
 - Classes: sports, politics, entertainment
- Probability that a document d belongs to class c
- Probability of class c given document d

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- Compute for every class

Naive Bayes

- Given a document d , what class does it belong to?
 - Find the most likely class c_{pred}

$$\begin{aligned}c_{\text{pred}} &= \arg \max_c P(c | d) \\&= \arg \max_c \frac{P(c)P(d | c)}{P(d)} \\&= \arg \max_c P(c)P(d | c)\end{aligned}$$

$$\begin{aligned}c_{pred} &= \arg \max P(c | d) \\&= \arg \max_c P(c)P(d | c)\end{aligned}$$

- How do we estimate $P(c)$?
- How do we estimate $P(d | c)$?
 - Naive Bayes assumption: words are independent
 - If document d is L words long

$$P(d | c) = P(w_1 | c)P(w_2 | c) \dots P(w_L | c)$$

Spam Filtering with Naive Bayes Classification

- Users create labeled data for free by tagging their own email, so training data is abundant!

SPAM click for pharmacy

OK free time today

SPAM online pharmacy link

OK no free time

OK free good pharmacy

SPAM pharmacy free link

OK for time today

OK time is money

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size:

click	for
pharmacy	free
time	today
online	link
no	good
is	money

12

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12
 $P(\text{spam}) =$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	Maximum Likelihood
OK	no free time	estimate
OK	free good pharmacy	
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12

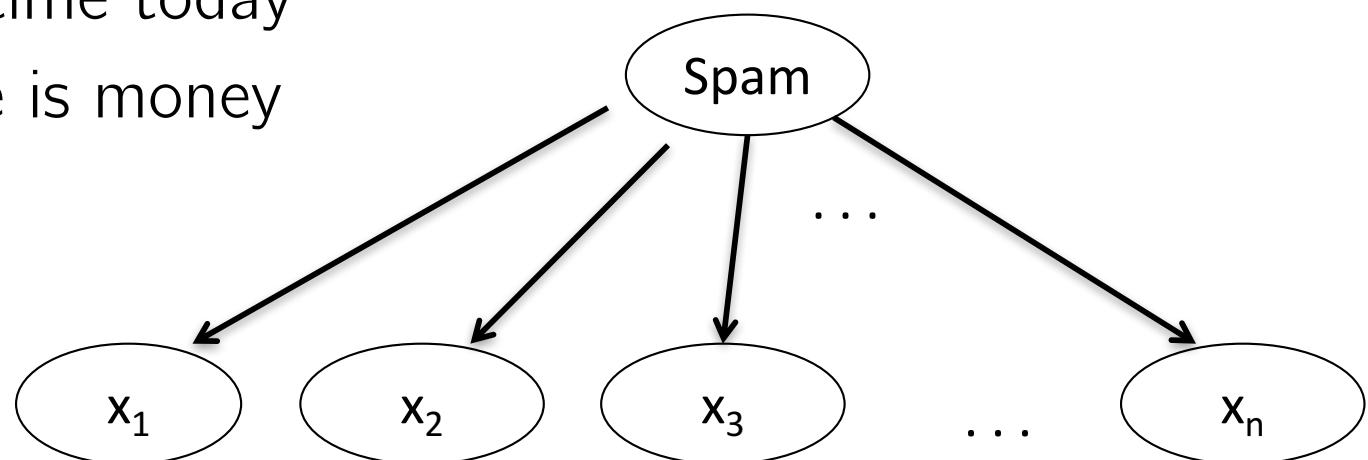
$$P(\text{spam}) = 3/8$$

$$P(\neg \text{spam}) = 5/8$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \mid \text{spam}) =$
OK	free good pharmacy	
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) =$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	



SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) =$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(\text{Msg} | \text{spam})}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for **pharmacy**”

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(w_1 | \text{spam}) P(w_2 | \text{spam}) P(w_3 | \text{spam})}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for **pharmacy**”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} \quad P(\neg\text{spam} | \text{Msg}) = \frac{P(\neg\text{spam}) P(\text{Msg} | \neg\text{spam})}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} \quad P(\neg\text{spam} | \text{Msg}) = \frac{5/8 \ 1/15 \ 1/15 \ 1/15}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} \quad P(\neg\text{spam} | \text{Msg}) = \frac{1/5400}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} > P(\neg\text{spam} | \text{Msg}) = \frac{1/5400}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for **pharmacy**”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} > P(\neg\text{spam} | \text{Msg}) = \frac{1/5400}{P(\text{Msg})}$$

Naive Bayes Classification, Evaluation

UC Davis LIN 127
Spring 2019

Kenji Sagae

Spam Filtering with Naive Bayes Classification

- Users create labeled data for free by tagging their own email, so training data is abundant!

SPAM click for pharmacy

OK free time today

SPAM online pharmacy link

OK no free time

OK free good pharmacy

SPAM pharmacy free link

OK for time today

OK time is money

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12
 $P(\text{spam}) =$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	Maximum Likelihood
OK	no free time	estimate
OK	free good pharmacy	
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12

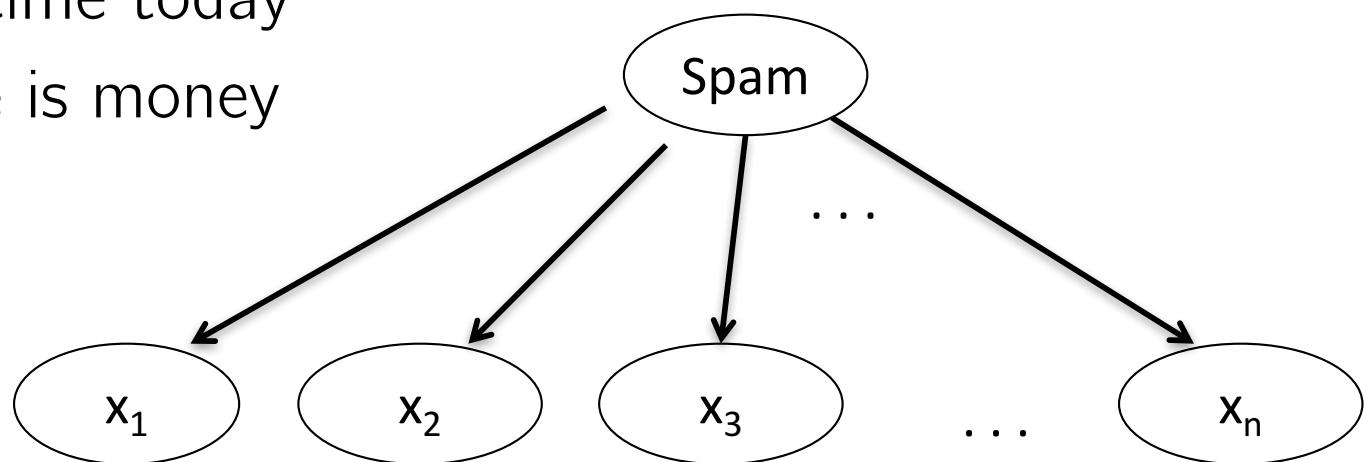
$$P(\text{spam}) = 3/8$$

$$P(\neg \text{spam}) = 5/8$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \mid \text{spam}) =$
OK	free good pharmacy	
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) =$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	



SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) =$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(\text{Msg} | \text{spam})}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \mid \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \mid \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for **pharmacy**”

$$P(\text{spam} \mid \text{Msg}) = \frac{P(\text{spam}) P(w_1 \mid \text{spam}) P(w_2 \mid \text{spam}) P(w_3 \mid \text{spam})}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} \quad P(\neg\text{spam} | \text{Msg}) = \frac{P(\neg\text{spam}) P(\text{Msg} | \neg\text{spam})}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"pharmacy for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} \quad P(\neg\text{spam} | \text{Msg}) = \frac{5/8 \ 1/15 \ 1/15 \ 1/15}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"pharmacy for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} \quad P(\neg\text{spam} | \text{Msg}) = \frac{1/5400}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for pharmacy”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} > P(\neg\text{spam} | \text{Msg}) = \frac{1/5400}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “pharmacy for **pharmacy**”

$$P(\text{spam} | \text{Msg}) = \frac{1/216}{P(\text{Msg})} > P(\neg\text{spam} | \text{Msg}) = \frac{1/5400}{P(\text{Msg})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

What happens if Msg = “time for pharmacy”?

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"time for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(\text{Msg} | \text{spam})}{P(\text{spam}) P(\text{Msg} | \text{spam}) + P(\neg\text{spam}) P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"time for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam}) P(\text{Msg} | \text{spam})}{P(\text{spam}) P(\text{Msg} | \text{spam}) + P(\neg\text{spam}) P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “time for **pharmacy**”

$$P(\text{spam} | \text{Msg}) = \frac{P(\text{spam})}{P(\text{spam})P(\text{Msg} | \text{spam}) + P(\neg\text{spam})P(\text{Msg} | \neg\text{spam})}$$

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

Msg = “time for pharmacy”

$$P(\text{spam} | \text{Msg}) = 0$$

Is this classification good?

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \text{spam}) = 1/3$
OK	free good pharmacy	$P(\text{pharmacy} \neg\text{spam}) = 1/15$
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

$\text{Msg} = \text{"time for pharmacy"}$

$$P(\text{spam} | \text{Msg}) = 0$$

We need smoothing!

e.g. add-one smoothing

Add-one smoothing

- Add one to the count of every observable event
- Suppose X has k possible outcomes, with counts n_1, n_2, \dots, n_k , which sum to N .
- Without smoothing: $P(X = i) = \frac{n_i}{N}$
 - Maximum likelihood estimate
- With add-one smoothing: $P(X = i) = \frac{n_i + 1}{N + k}$
- Not great, but very simple

- Computing $P(c)$
 - Maximum likelihood estimate
 - Number of documents of class c divided by total number of documents
- Computing $P(w | c)$
 - Number of times word w appears in documents in class c , divided by the total number of words in documents in class c
 - With add-one smoothing

$$P(w | c) = \frac{n + 1}{N + k}$$

where n is the number of times w appears in c , N is the total number of words in c , and k is the size of the vocabulary

SPAM	click for pharmacy	Vocabulary size: 12
OK	free time today	$P(\text{spam}) = 3/8$
SPAM	online pharmacy link	$P(\neg\text{spam}) = 5/8$
OK	no free time	$P(\text{pharmacy} \mid \text{spam}) =$
OK	free good pharmacy	
SPAM	pharmacy free link	
OK	for time today	
OK	time is money	

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12

$$P(\text{spam}) = 3/8$$

$$P(\neg \text{spam}) = 5/8$$

$$P(\text{pharmacy} | \text{spam}) =$$

$$= (3+1) / (9+12)$$

$$= 4 / 21$$

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12

$$P(\text{spam}) = 3/8$$

$$P(\neg \text{spam}) = 5/8$$

$$P(\text{pharmacy} | \text{spam}) =$$

$$= (3+1) / (9+12)$$

$$= 4 / 21$$

$$P(\text{time} | \text{spam}) =$$

SPAM	click for pharmacy
OK	free time today
SPAM	online pharmacy link
OK	no free time
OK	free good pharmacy
SPAM	pharmacy free link
OK	for time today
OK	time is money

Vocabulary size: 12

$$P(\text{spam}) = 3/8$$

$$P(\neg \text{spam}) = 5/8$$

$$P(\text{pharmacy} | \text{spam}) =$$

$$= (3+1) / (9+12)$$

$$= 4 / 21$$

$$P(\text{time} | \text{spam}) = 1 / 21$$

Beyond bag-of-words

- Many kinds of features are possible
 - Not limited to words
- User is addressed by the correct name (binary)
- Sender is someone you have sent email to before
- ...
- But in a Naive Bayes model we have the usual independence assumption
 - There are many other classification approaches
- What about stop words, stemmers, etc.
 - Trust the data

Evaluation

- Simple accuracy
 - How many predictions did I get right?
 - Number of correct predictions / Total number of predictions
 - Why is this sometimes not a good way to evaluate a classifier?
- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

accuracy =

HAM	HAM
-----	-----

HAM	SPAM
-----	------

HAM	HAM
-----	-----

HAM	SPAM
-----	------

SPAM	HAM
------	-----

Evaluation

- Simple accuracy
 - How many predictions did I get right?
 - Number of correct predictions / Total number of predictions
 - Why is this sometimes not a good way to evaluate a classifier?
- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

accuracy = 3 / 6

HAM	HAM
-----	-----

HAM	SPAM
-----	------

HAM	HAM
-----	-----

HAM	SPAM
-----	------

SPAM	HAM
------	-----

Evaluation

- Simple accuracy
 - How many predictions did I get right?
 - Number of correct predictions / Total number of predictions
 - Why is this sometimes not a good way to evaluate a classifier?
- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

accuracy = 3 / 6 = 50%

HAM	HAM
-----	-----

HAM	SPAM
-----	------

HAM	HAM
-----	-----

HAM	SPAM
-----	------

SPAM	HAM
------	-----

Evaluation

- Precision of class c
 - Out of everything predicted to be of class c, what fraction was *actually* of class c?
(fraction of documents classified as c that I got right)

$$precision_c = \frac{count(\text{correctly_classified_as_}c)}{count(\text{classified_as_}c)}$$

- Recall of class c
 - Out of everything in class c, what fraction did I find?
(fraction of documents in class c that I got right)

$$recall_c = \frac{count(\text{correctly_classified_as_}c)}{count(\text{belongs_in_}c)}$$

Classification, Evaluation

UC Davis LIN 127
Spring 2019

Kenji Sagae

Evaluation

- Simple accuracy
 - How many predictions did I get right?
 - Number of correct predictions / Total number of predictions
 - Why is this sometimes not a good way to evaluate a classifier?
- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

accuracy =

HAM	HAM
-----	-----

HAM	SPAM
-----	------

HAM	HAM
-----	-----

HAM	SPAM
-----	------

SPAM	HAM
------	-----

Evaluation

- Simple accuracy
 - How many predictions did I get right?
 - Number of correct predictions / Total number of predictions
 - Why is this sometimes not a good way to evaluate a classifier?
- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

accuracy = 3 / 6

HAM	HAM
-----	-----

HAM	SPAM
-----	------

HAM	HAM
-----	-----

HAM	SPAM
-----	------

SPAM	HAM
------	-----

Evaluation

- Simple accuracy
 - How many predictions did I get right?
 - Number of correct predictions / Total number of predictions
 - Why is this sometimes not a good way to evaluate a classifier?
- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

accuracy = 3 / 6 = 50%

HAM	HAM
-----	-----

HAM	SPAM
-----	------

HAM	HAM
-----	-----

HAM	SPAM
-----	------

SPAM	HAM
------	-----

Evaluation

- Precision of class c
 - Out of everything predicted to be of class c, what fraction was *actually* of class c?
(fraction of documents classified as c that I got right)

$$precision_c = \frac{count(\text{correctly_classified_as_}c)}{count(\text{classified_as_}c)}$$

- Recall of class c
 - Out of everything in class c, what fraction did I find?
(fraction of documents in class c that I got right)

$$recall_c = \frac{count(\text{correctly_classified_as_}c)}{count(\text{belongs_in_}c)}$$

Evaluation

- F-score: combining precision and recall

$$F_1 = \frac{2 \cdot \textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}$$

- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

HAM	HAM
-----	-----

HAM	SPAM
-----	------

HAM	HAM
-----	-----

SPAM	HAM
------	-----

HAM	SPAM
-----	------

Evaluation

- F-score: combining precision and recall

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

precision = 1/3 = 0.33

HAM	HAM
-----	-----

HAM	SPAM
-----	------

HAM	HAM
-----	-----

SPAM	HAM
------	-----

HAM	SPAM
-----	------

Evaluation

- F-score: combining precision and recall

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

precision = 1/3 = 0.33

HAM	HAM
-----	-----

HAM	SPAM
-----	------

recall = 1/2 = 0.5

HAM	HAM
-----	-----

SPAM	HAM
------	-----

HAM	SPAM
-----	------

Evaluation

- F-score: combining precision and recall

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- Example:

<u>Actual</u>	<u>Predicted</u>
---------------	------------------

SPAM	SPAM
------	------

precision = 1/3 = 0.33

HAM	HAM
-----	-----

HAM	SPAM
-----	------

recall = 1/2 = 0.5

HAM	HAM
-----	-----

SPAM	HAM
------	-----

$$F_1 = 2(0.5)(0.33)/(0.5+0.33)$$

HAM	SPAM
-----	------

$$= 0.33/0.83 = 0.4$$

Training, Development, Evaluation

- What is a training set?
- What is a development set?
- What is a test set?

HW 3

UC Davis LIN 127
Spring 2019

Kenji Sagae

HW 3: Text Classification (part 2)

- Write a python program that reads your sentiment model (sentiment.nb) and classifies the test set
- For each file in the test set
 - Write: file name + ' ' + 'pos' if the review is positive
 - Write file name + ' ' + 'neg' if the review is negative
- Turn in
 - Your program, nbtest.py
 - The output of your program, sentiment.out
 - Optional: any comments you may have as a separate file
 - Comment your code
 - Any line that begins with # is a comment, not python code
- Use NLTK, but no packages that are not in the Standard Python Library (e.g. no scipy, pandas, etc.)

```
# restore model from sentiment.nb
with open('sentiment.nb', 'rb') as f:
    model = pickle.load(f)

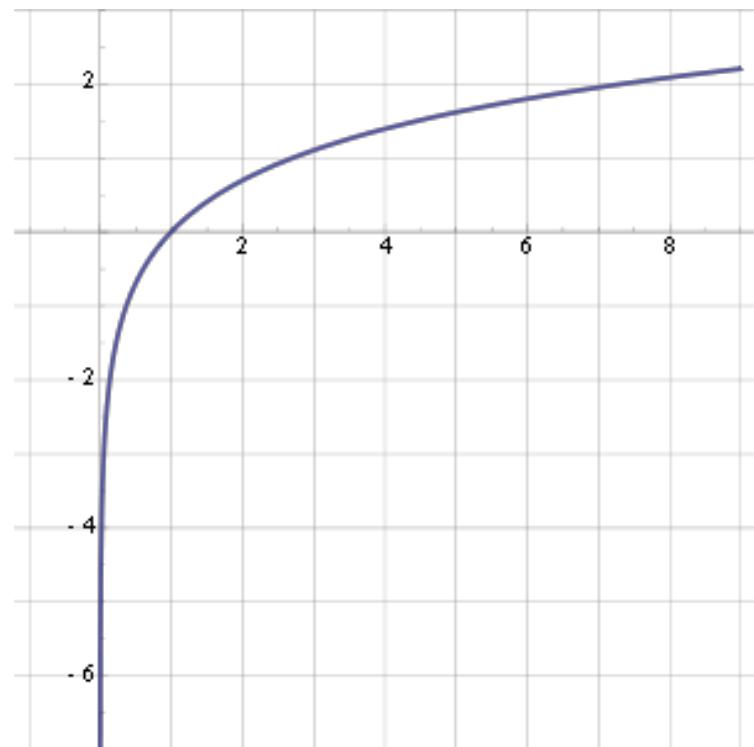
# number of tokens in pos freq distribution
pos_count = model['pos_fd'].N()

# number of types in pos freq distribution
pos_v = model['pos_fd'].B()

# p("thanks" | pos), without add-one smoothing
cp = model['pos_fd']['thanks'] / pos_count

# log
# only need to import math once, not every
time!
import math
log_cp = math.log(cp)
```

- What happens if the number of tokens is large?
- Some probabilities can get very small
 - Especially $P(\text{rare_word} \mid c)$
- Underflow!
- $\log(A) < \log(B)$ for $0 < A < B$
- $\log(A \times B) = \log(A) + \log(B)$



- What does the following program do?

```
values = [2,4,5,7,9]
```

```
sum = 0
```

```
for v in values:
```

```
    sum = sum + v
```

- Assuming we have a frequency distribution called pos_fd, what does the following program do?

```
words = ['the', 'dog', 'eats']
```

```
sum = 0
```

```
for w in words:
```

```
    w_count = pos_fd[w]
```

```
    sum = sum + w_count
```

Projects, Sample Application

UC Davis LIN 127
Spring 2019

Kenji Sagae

```
# restore model from sentiment.nb
with open('sentiment.nb', 'rb') as f:
    model = pickle.load(f)

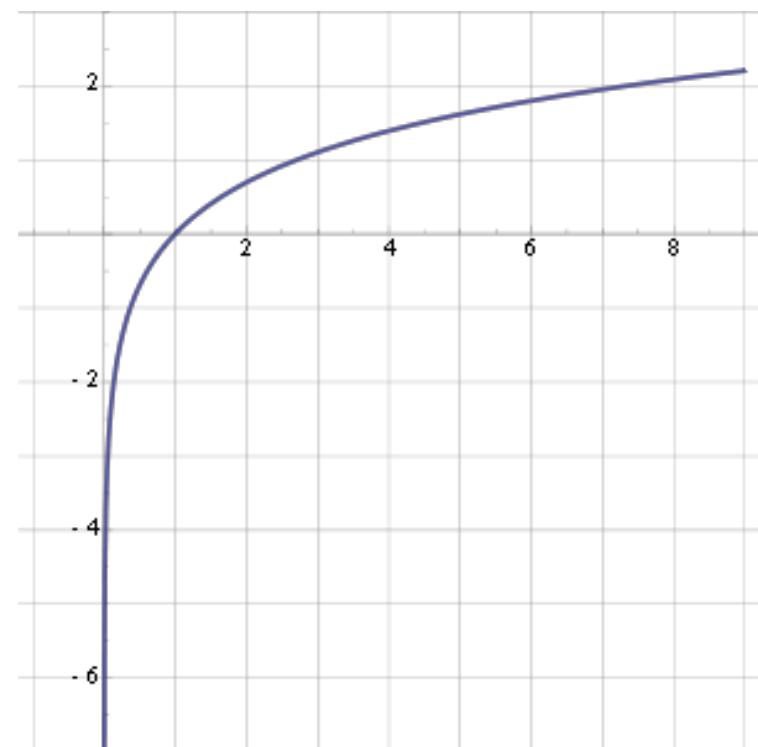
# number of tokens in pos freq distribution
pos_count = model['pos_fd'].N()

# number of types in pos freq distribution
pos_v = model['pos_fd'].B()

# p("thanks" | pos), without add-one smoothing
cp = model['pos_fd']['thanks'] / pos_count

# log
# only need to import math once, not every
time!
import math
log_cp = math.log(cp)
```

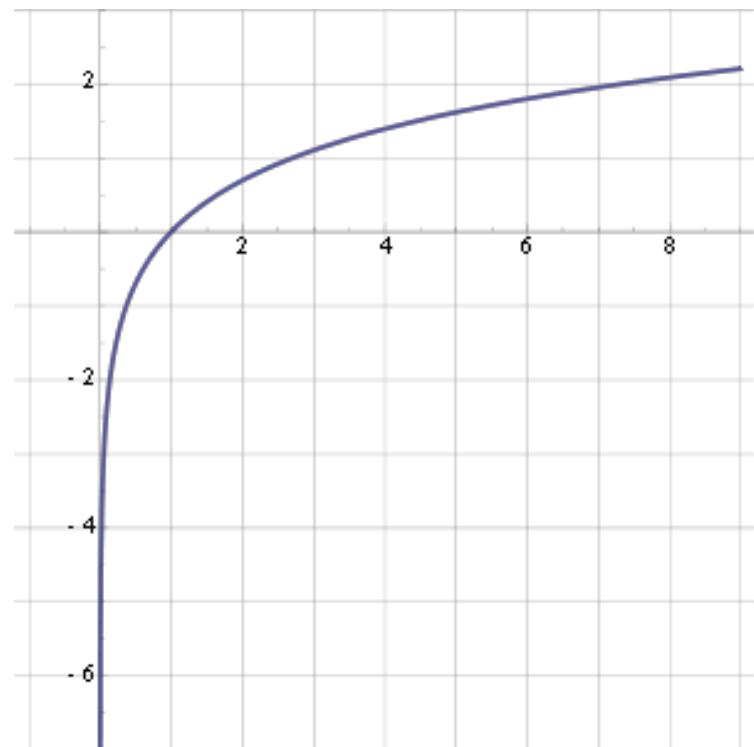
- What happens if the number of tokens is large?
- Some probabilities can get very small
 - Especially $P(\text{rare_word} \mid c)$
- Underflow!
- $\log(A) < \log(B)$ for $0 < A < B$
- $\log(A \times B) = \log(A) + \log(B)$



Project

- Two weeks, three students
 - Total scope, work: approx. 3x HW3
 - Make sure there is enough distinct work for everyone
 - Contributions must be specified explicitly
- Report due at the end of instruction
- Must focus on language
 - Not programming, not applications, etc.

- What happens if the vocabulary is large?
- Some probabilities can get very small
 - Especially $P(\text{rare_word} \mid c)$
- Underflow!
- $\log(A) < \log(B)$ for $0 < A < B$
- $\log(A \times B) = \log(A) + \log(B)$



- What does the following program do?

```
base = 2
```

```
exponent = 3
```

```
result = 1
```

```
while exponent > 0:
```

```
    result = result * base
```

```
    exponent = exponent - 1
```

- What does the following program do?

```
values = [2,4,5,7,9]
```

```
sum = 0
```

```
for v in values:
```

```
    sum = sum + v
```

- What does the following program do?

```
words = ['the', 'dog', 'eats']
```

```
sum = 0
```

```
for w in words:
```

```
    w_count = ham_fd[w]
```

```
    sum = sum + w_count
```

Sample application:
Cross-cultural analysis
of personal narratives

Personal Narratives

- First-person account of events
 - Examples: diary entries, personal blog posts
 - Typically include *objective* and *subjective* statements
 - Often intended for more than simply relating events
 - Influence opinion, elicit empathy, entertain, etc.
- The narrator is also a participant in the story
 - Unique advantage in influencing interpretation
 - The narrator exists in two *narrative levels*
 - As storyteller at the time of narration
 - As a participant in events in the past
- Corpus: personal narratives from social media content

Narrative Levels

- In narratology, also known as *diegetic levels*
 - Genette, 1980
- (Intra)Diegetic level: the universe of the story, where events take place and participants exist
- Extradiegetic level: where the act of narration takes place, where the narrator speaks to the reader



Example: car chase

Images: diegetic

Sound of cars, characters talking, etc: diegetic

Soundtrack: extradiegetic



Narrative Levels and subjectivity

- Some text refers only to the diegetic level
 - Description of events in the past
 - Thoughts and emotions experienced during these events
- Some text does *not* refer to the diegetic level
 - Background and values that frame past events
 - Usually meant to influence reader's interpretation
- Statements referring to either level can be *objective* or *subjective*
 - Subjectivity can be expressed at either narrative level

Personal Narrative Example

- Real blog post about a nurse who crossed a picket line after reading about the effect of a nurse's strike on patient care

(typos left unedited)

I often jokingly identify myself as being like the pregnant cop from the movie Fargo. You know the one with the thick Midwestern accent and the kinky hair. The humor comes from the incredibly real similarities we share. I've lived in Minnesota my whole life. I'm a middle aged version of the girl next door; your quintessential "good girl". I've always followed the rules and done my best not to upset anybody.

There had been rumors of a possible nursing strike here in Minnesota for quite a while before that day I read the article that mentioned the statistics from the strike in New York. To be honest, even though I am a nurse myself, I hadn't really paid much attention, until I read that hospitals see a 19.4 percent increase in patient death during a nursing strike. It hit me like a baseball bat to the shin, and I couldn't stop thinking about the number for days.

I decided I couldn't sit back and watch the effects of this strike play out, in real time, here in my city. For once in my life I did what felt right, even though some people thought it was incredibly wrong. I crossed the picket line and went into the hospital to work and to help the patients during the strike.

I often jokingly identify myself as being like the pregnant cop from the movie Fargo. You know the one with the thick Midwestern accent and the kinky hair. The humor comes from the incredibly real similarities we share. I've lived in Minnesota my whole life. I'm a middle aged version of the girl next door; your quintessential "good girl". I've always followed the rules and done my best not to upset anybody.

There had been rumors of a possible nursing strike here in Minnesota for quite a while before that day I read the article that mentioned the statistics from the strike in New York. To be honest, even though I am a nurse myself, I hadn't really paid much attention, until I read that hospitals see a 19.4 percent increase in patient death during a nursing strike. It hit me like a baseball bat to the shin, and I couldn't stop thinking about the number for days.

I decided I couldn't sit back and watch the effects of this strike play out, in real time, here in my city. For once in my life I did what felt right, even though some people thought it was incredibly wrong. I crossed the picket line and went into the hospital to work and to help the patients during the strike.

I often jokingly identify myself as being like the pregnant cop from the movie Fargo. You know the one with the thick Midwestern accent and the kinky hair. The humor comes from the incredibly real similarities we share. I've lived in Minnesota my whole life. I'm a middle aged version of the girl next door; your quintessential "good girl". I've always followed the rules and done my best not to upset anybody.

There had been rumors of a possible nursing strike here in Minnesota for quite a while before that day I read the article that mentioned the statistics from the strike in New York. To be honest, even though I am a nurse myself, I hadn't really paid much attention, until I read that hospitals see a 19.4 percent increase in patient death during a nursing strike. It hit me like a baseball bat to the shin, and I couldn't stop thinking about the number for days.

I decided I couldn't sit back and watch the effects of this strike play out, in real time, here in my city. For once in my life I did what felt right, even though some people thought it was incredibly wrong. I crossed the picket line and went into the hospital to work and to help the patients during the strike.

I drove to the unfamiliar hospital with bated breath this morning. Doing things in the dark always makes them seem more naughty, so as I headed off at 4am though the night, I felt a tinge of excitement mixed along with my nervousness and anticipation. You don't get much job training when you're a "scab". They just throw you into the pool and see if you swim..... I swam; not only for the patients, but for myself.

The patients, doctors, and other staff were so welcoming and thankful to us for stepping outside our normal routine and into theirs. They showed us where to get coffee, held warm smiles on their faces, and some of them even volleyed around words like "hero" and "admirable". Oh, and the other "scabs"...they were good girls (and boys) just like me, who laughed about being called "heros".

I'm no hero. I'm a klutz. I'm a dork. I may be a big nerd, and you may not agree with what I did, but I feel in my heart that I finally did something right, and it felt good!

I drove to the unfamiliar hospital with bated breath this morning. Doing things in the dark always makes them seem more naughty, so as I headed off at 4am though the night, I felt a tinge of excitement mixed along with my nervousness and anticipation. You don't get much job training when you're a "scab". They just throw you into the pool and see if you swim..... I swam; not only for the patients, but for myself.

The patients, doctors, and other staff were so welcoming and thankful to us for stepping outside our normal routine and into theirs. They showed us where to get coffee, held warm smiles on their faces, and some of them even volleyed around words like "hero" and "admirable". Oh, and the other "scabs"...they were good girls (and boys) just like me, who laughed about being called "heros".

I'm no hero. I'm a klutz. I'm a dork. I may be a big nerd, and you may not agree with what I did, but I feel in my heart that I finally did something right, and it felt good!

I drove to the unfamiliar hospital with bated breath this morning. Doing things in the dark always makes them seem more naughty, so as I headed off at 4am though the night, I felt a tinge of excitement mixed along with my nervousness and anticipation. You don't get much job training when you're a "scab". They just throw you into the pool and see if you swim..... I swam; not only for the patients, but for myself.

The patients, doctors, and other staff were so welcoming and thankful to us for stepping outside our normal routine and into theirs. They showed us where to get coffee, held warm smiles on their faces, and some of them even volleyed around words like "hero" and "admirable". Oh, and the other "scabs"...they were good girls (and boys) just like me, who laughed about being called "heros".

I'm no hero. I'm a klutz. I'm a dork. I may be a big nerd, and you may not agree with what I did, but I feel in my heart that I finally did something right, and it felt good!

I drove to the unfamiliar hospital with bated breath this morning. Doing things in the dark always makes them seem more naughty, so as I headed off at 4am though the night, I felt a tinge of excitement mixed along with my nervousness and anticipation. You don't get much job training when you're a "scab". They just throw you into the pool and see if you swim..... I swam; not only for the patients, but for myself.

The patients, doctors, and other staff were so welcoming and thankful to us for stepping outside our normal routine and into theirs. They showed us where to get coffee, held warm smiles on their faces, and some of them even volleyed around words like "hero" and "admirable". Oh, and the other "scabs"...they were good girls (and boys) just like me, who laughed about being called "heros".

I'm no hero. I'm a klutz. I'm a dork. I may be a big nerd, and you may not agree with what I did, but I feel in my heart that I finally did something right, and it felt good!

I drove to the unfamiliar hospital with bated breath this morning. Doing things in the dark always makes them seem more naughty, so as I headed off at 4am though the night, I felt a tinge of excitement mixed along with my nervousness and anticipation. You don't get much job training when you're a "scab". They just throw you into the pool and see if you swim..... I swam; not only for the patients, but for myself.

The patients, doctors, and other staff were so welcoming and thankful to us for stepping outside our normal routine and into theirs. They showed us where to get coffee, held warm smiles on their faces, and some of them even volleyed around words like "hero" and "admirable". Oh, and the other "scabs"...they were good girls (and boys) just like me, who laughed about being called "heros".

I'm no hero. I'm a klutz. I'm a dork. I may be a big nerd, and you may not agree with what I did, but I feel in my heart that I finally did something right, and it felt good!

I drove to the unfamiliar hospital with bated breath this morning. Doing things in the dark always makes them seem more naughty, so as I headed off at 4am though the night, I felt a tinge of excitement mixed along with my nervousness and anticipation. You don't get much job training when you're a "scab". They just throw you into the pool and see if you swim..... I swam; not only for the patients, but for myself.

The patients, doctors, and other staff were so welcoming and thankful to us for stepping outside our normal routine and into theirs. They showed us where to get coffee, held warm smiles on their faces, and some of them even volleyed around words like "hero" and "admirable". Oh, and the other "scabs"...they were good girls (and boys) just like me, who laughed about being called "heros".

I'm no hero. I'm a klutz. I'm a dork. I may be a big nerd, and you may not agree with what I did, but I feel in my heart that I finally did something right, and it felt good!

I drove to the unfamiliar hospital with bated breath this morning. Doing things in the dark always makes them seem more naughty, so as I headed off at 4am though the night, I felt a tinge of excitement mixed along with my nervousness and anticipation. You don't get much job training when you're a "scab". They just throw you into the pool and see if you swim..... I swam; not only for the patients, but for myself.

The patients, doctors, and other staff were so welcoming and thankful to us for stepping outside our normal routine and into theirs. They showed us where to get coffee, held warm smiles on their faces, and some of them even volleyed around words like "hero" and "admirable". Oh, and the other "scabs"...they were good girls (and boys) just like me, who laughed about being called "heros".

I'm no hero. I'm a klutz. I'm a dork. I may be a big nerd, and you may not agree with what I did, but I feel in my heart that I finally did something right, and it felt good!

- Some text refers only to the diegetic level
 - Description of events in the past
 - Thoughts and emotions experienced during these events
- Some text does *not* refer to the diegetic level
 - Background and values that frame past events
 - Usually meant to influence reader's interpretation
- Statements referring to either level can be *objective* or *subjective*

Subjectivity in Text

- Subjective language: expression of *private states*
 - Following Wiebe et al. (2004)
- Private states: emotions, opinions, beliefs, thoughts, goals, judgments
 - Not open to external observation or verification
- Closely related to NLP work on sentiment classification
 - Typically polarity of reviews (e.g. Pang and Lee, 2004)

Subjectivity in Personal Narratives

- No polarity classification
 - Not positive vs. negative
 - Objective vs. subjective
- Narrator expresses thoughts, emotions, etc.
 - As a character, at the diegetic level
 - As the storyteller, at the extradiegetic level

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Late one night my three-year-old son came into our room uncontrollably crying. This kid was obviously scared out of his mind and was completely hysterical. He told me that he could not sleep because there were monsters under his bed.

I felt really bad that he was so scared, so I let him stay in our bed for a few minutes to calm down, but when it was time to go back to his own bed he was still afraid of the monsters. Finally, I made up a lie to make him feel better. I told him that I had a special blanket that keeps monsters away.

With his monster-repelling blanket, he was able to go back to sleep. I know that usually lying to my child is wrong, but as mothers sometimes we have to do it to protect our children and ensure that they feel safe.

Data Collection and Annotation

- Corpus-driven analysis and modeling of narratives requires data
 - Narrative text, annotated with targeted characteristics
 - English (US), Farsi (Iran), and Mandarin (China)
- Manual annotation of \sim 1,200 personal stories (\sim 40,000 discourse segments)
 - Posts involving socially questionable behavior
 - English, Farsi, Mandarin
 - Multiple annotators in each language

Experiment

classification using 600 blog posts

- Dataset: about 600 narratives from blog posts in English
 - About 20,000 discourse segments
 - 100 narratives (3,000 segments) held-out for testing
- Segmentation accuracy: 84%
- Classification of narrative level: 91%
- Classification of subjectivity: 72%
- Similar results with Mandarin and Farsi narratives

Mundane narratives vs value-laden narratives

	Mundane topics	Moral decisions
Avg. #segments	38	38
Extradiegetic	40%	41%
Subjective	65%	70%
Subjective extradiegetic	80%	86%
Subjective diegetic	55%	60%

Cross-cultural comparison

	Mandarin	English	Farsi
# Segments	34	38	35
Subjectivity	65%	65%	68%
Extradiegetic	49%	40%	32%
Subjective extradiegetic	73%	80%	92%
Subjective diegetic	55%	55%	58%

Annotated Corpora: Parts-of-Speech

UC Davis LIN 127
Spring 2019

Kenji Sagae

Annotated Corpora

- Corpora can be plain text
 - Sometimes with labels for documents
- Corpora can be annotated at several levels of analysis
 - Tokenization
 - Morphological analysis
 - Word categories
 - Sentence categories/labels
 - Etc.

Manual Annotation

- Start with plain text
- Apply lower level annotations or preprocessing
 - E.g. Tokenization
- Annotate manually
 - Requires clear guidelines (annotation manual)
 - How do we know the quality of the guidelines
 - Often requires trained annotators
- Annotator agreement is task dependent

Parts of Speech

- Linguistic categories for words
 - word class, lexical category
 - noun, verb, article, adjective, adverb, preposition...
- Open class
 - noun, verb, adjective, adverb
 - content-bearing, refer to objects, actions, etc.
 - new members can be added
- Closed class
 - preposition, pronoun, article, conjunction
 - typically functional

Part-of-Speech Tags

(Example English tagset)

- Nouns
 - NN (common), NNS (plural), NNP (proper), NNPS
- Verbs
 - VB (base), VBP (present), VBZ (present, 3rd person singular),
VBD (past), VBN (past participle), VBG (gerund)
 - What is the difference between VB and VBP?
 - *I eat* vs *I have to eat*
- Adjectives
 - JJ, JJR (comparative), JJS (superlative)
- Adverbs
 - RB, RBR (comparative), RBS (superlative)

Part-of-Speech Tags

(Example English tagset, continued)

- CC (conjunction)
- CD (number)
- DT (determiner)
- EX (existential *there*)
- FW (foreign word)
- IN (preposition or
subordinating conjunction)
- MD (modal)
- PP (personal pronoun)
- PP\$ (possessive pronoun)
- RP (particle)
- SYM (symbol)
- TO (to)
- UH (interjection)
- WDT (wh-determiner)
- WP (wh-pronoun)
- WP\$ (possessive wh-
pronoun)
- WRB (wh-adverb)

Part-of-Speech Ambiguity

- What is the part of speech of:
 - the
 - bank
 - house
 - back
 - object
 - run
 - objective

Example: Adjectives (JJ)

- JJs often end in *ing* and can be confused with VBG
 - For example: interesting, depressing, striking
- Class specific tests are often useful
 - Is it gradable? (intensifier, comparative)
 - Her talk was very interesting
 - Her talk was more interesting than others
 - Can it be negated by the prefix *un*?
 - uninteresting
 - Be/become
 - The conversation *became* depressing
- Not all JJ tests apply to all JJs!
 - *undepressing

Part of Speech Ambiguity

- Squad helps dog bite victim

Part of Speech Ambiguity

- Squad helps dog bite victim
- NN VBZ NN ?? NN

Part of Speech Ambiguity

- Squad helps dog bite victim
- NN VBZ NN NN NN

Part of Speech Ambiguity

- Squad helps dog bite victim
- NN VBZ NN VB NN

Part-of-Speech Tagging

- Determine the POS tag for each word in a sentence

NN VBZ NN NN NN

- Squad helps dog bite victim

Part-of-Speech Tagging

- Roughly: determine the POS tag for each word in a sentence

NN VBZ NN NN NN

- Squad helps dog bite victim

- Useful

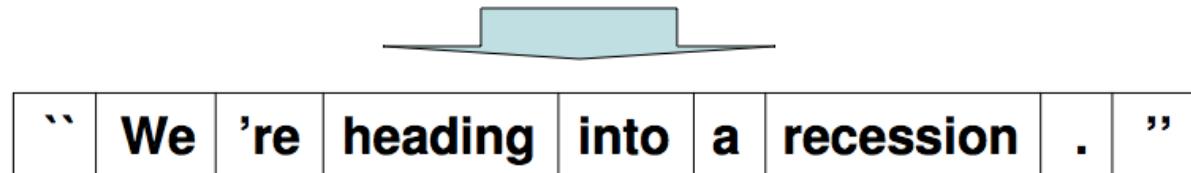
- Speech synthesis (TTS): lead, record, object
- Input for parser (as part of a pipeline)
- Find patterns
 - e.g. base noun phrases, named entities
- Back-off when words are too sparse

Part-of-speech tagging

- Learning from data very effective
- A sequence labeling task
 - Given a sequence of tokens $w_1 w_2 w_3 \dots w_n$, find one tag t_i for each token w_i
 - Find a sequence of tags $t_1 t_2 t_3 \dots t_n$
 - In POS tagging, each tag corresponds to a part-of-speech
- High accuracy with supervised learning
 - Needs manually annotated training data
- Sometimes attempted with unsupervised learning
 - Learn from text alone (no annotations)
 - We will not cover this

- Assign one tag per word
 - What is a *word*? Should really be *token*.
 - Sometimes not what you would expect
 - Split at whitespace?
 - Can work for English bag-of-words, but not quite enough of POS tagging
- Tokenization
 - Split a sentence into tokens

“We’re heading into a recession.”



Tokenization

- A relatively straightforward task in English
 - More challenging in languages with no whitespace
 - Even a straightforward task in NLP can be full of uncertainty and many possible solutions
- Often whitespace + list of special cases
 - Separate possessive 's, separate contractions
 - Mary's -> Mary 's
 - You're -> You 're
 - Separate punctuation, parenthesis, quotation marks
 - Hello. -> Hello .
 - "Hello" -> " Hello "

Tokenization in practice (English)

- Usually a short list of regular expressions (20 or so)
 - Example sed script here:
<http://www.cis.upenn.edu/~treebank/tokenization.html>
- Easy to do a decent job, impossible to do perfectly
- Tokens include words, punctuation, numbers, etc.
 - Each token gets one POS tag

Part-of-Speech Tagging

UC Davis LIN 127
Spring 2019

Kenji Sagae

Parts of Speech

- Linguistic categories for words
 - word class, lexical category
 - noun, verb, article, adjective, adverb, preposition...
- Open class
 - noun, verb, adjective, adverb
 - content-bearing, refer to objects, actions, etc.
 - new members can be added
- Closed class
 - preposition, pronoun, article, conjunction
 - typically functional

Part-of-Speech Tags

(Example English tagset)

- Nouns
 - NN (common), NNS (plural), NNP (proper), NNPS
- Verbs
 - VB (base), VBP (present), VBZ (present, 3rd person singular),
VBD (past), VBN (past participle), VBG (gerund)
 - What is the difference between VB and VBP?
 - *I eat* vs *I have to eat*
- Adjectives
 - JJ, JJR (comparative), JJS (superlative)
- Adverbs
 - RB, RBR (comparative), RBS (superlative)

Part-of-Speech Tags

(Example English tagset, continued)

- CC (conjunction)
- CD (number)
- DT (determiner)
- EX (existential *there*)
- FW (foreign word)
- IN (preposition or
subordinating conjunction)
- MD (modal)
- PP (personal pronoun)
- PP\$ (possessive pronoun)
- RP (particle)
- SYM (symbol)
- TO (to)
- UH (interjection)
- WDT (wh-determiner)
- WP (wh-pronoun)
- WP\$ (possessive wh-
pronoun)
- WRB (wh-adverb)

Part-of-Speech Ambiguity

- What is the part of speech of:
 - the
 - bank
 - house
 - back
 - object
 - run
 - objective

Part of Speech Ambiguity

- Squad helps dog bite victim

Part of Speech Ambiguity

- Squad helps dog bite victim
- NN VBZ NN ?? NN

Part of Speech Ambiguity

- Squad helps dog bite victim
- NN VBZ NN NN NN

Part of Speech Ambiguity

- Squad helps dog bite victim
- NN VBZ NN VB NN

Part-of-Speech Tagging

- Determine the POS tag for each word in a sentence

NN VBZ NN NN NN

- Squad helps dog bite victim

Part-of-Speech Tagging

- Roughly: determine the POS tag for each word in a sentence

NN VBZ NN NN NN

- Squad helps dog bite victim

- Useful

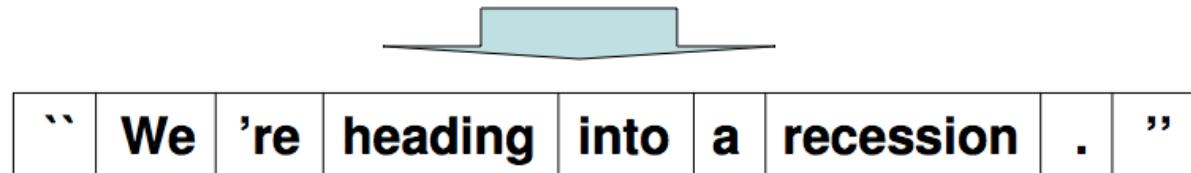
- Speech synthesis (TTS): lead, record, object
- Input for parser (as part of a pipeline)
- Find patterns
 - e.g. base noun phrases, named entities
- Back-off when words are too sparse

Part-of-speech tagging

- Learning from data very effective
- A sequence labeling task
 - Given a sequence of tokens $w_1 w_2 w_3 \dots w_n$, find one tag t_i for each token w_i
 - Find a sequence of tags $t_1 t_2 t_3 \dots t_n$
 - In POS tagging, each tag corresponds to a part-of-speech
- High accuracy with supervised learning
 - Needs manually annotated training data
- Sometimes attempted with unsupervised learning
 - Learn from text alone (no annotations)
 - We will not cover this

- Assign one tag per word
 - What is a *word*? Should really be *token*.
 - Sometimes not what you would expect
 - Split at whitespace?
 - Can work for English bag-of-words, but not quite enough of POS tagging
- Tokenization
 - Split a sentence into tokens

“We’re heading into a recession.”



Tokenization

- A relatively straightforward task in English
 - More challenging in languages with no whitespace
 - Even a straightforward task in NLP can be full of uncertainty and many possible solutions
- Often whitespace + list of special cases
 - Separate possessive 's, separate contractions
 - Mary's -> Mary 's
 - You're -> You 're
 - Separate punctuation, parenthesis, quotation marks
 - Hello. -> Hello .
 - "Hello" -> " Hello "

Tokenization in practice (English)

- Usually a short list of regular expressions (20 or so)
 - Short description with script here:
ftp://ftp.cis.upenn.edu/pub/treebank/public_html/tokenization.html
- Easy to do a decent job, impossible to do perfectly
- Tokens include words, punctuation, numbers, etc.
 - Each token gets one POS tag

Creating training data for POS tagging (manual annotation)

- Plenty of ambiguity and borderline cases
- How can we get reliable annotation?
 - Annotation that can be repeated by the same or different annotators, with the same results
- Careful guidelines
 - General principles
 - Tests
 - A lot of specific examples
 - A lot of special cases

Part-of-Speech Tagging

- One tag per token
- Accuracy: #correct tags / #words
- Current accuracy: ~97%
 - (people: ~98%)
- Training data: correctly tagged text
- Baseline: tag each word with its most frequent tag
 - 90% accurate!
 - Many words are unambiguous (the, a, she, ...)

POS Tagging with HMMs

- When tagging a word, we look at:
 - The word itself
 - The grammatical environment of the word
- We want the best sequence of tags T for a sentence S

POS Tagging with HMMs

- When tagging a word, we look at:
 - The word itself
 - The grammatical environment of the word
 - We want the best sequence of tags T for a sentence S
 - $\operatorname{argmax}_T p(T|S)$
- $$p(T|S) = \frac{p(S|T)p(T)}{p(S)}$$
- so: $\operatorname{argmax}_T p(T|S) = \operatorname{argmax}_T p(S|T)p(T)$

POS Tagging with HMMs

$$\arg \max_T p(T|S) = \arg \max_T p(S|T)p(T)$$

- We decompose $p(S|T)$ into

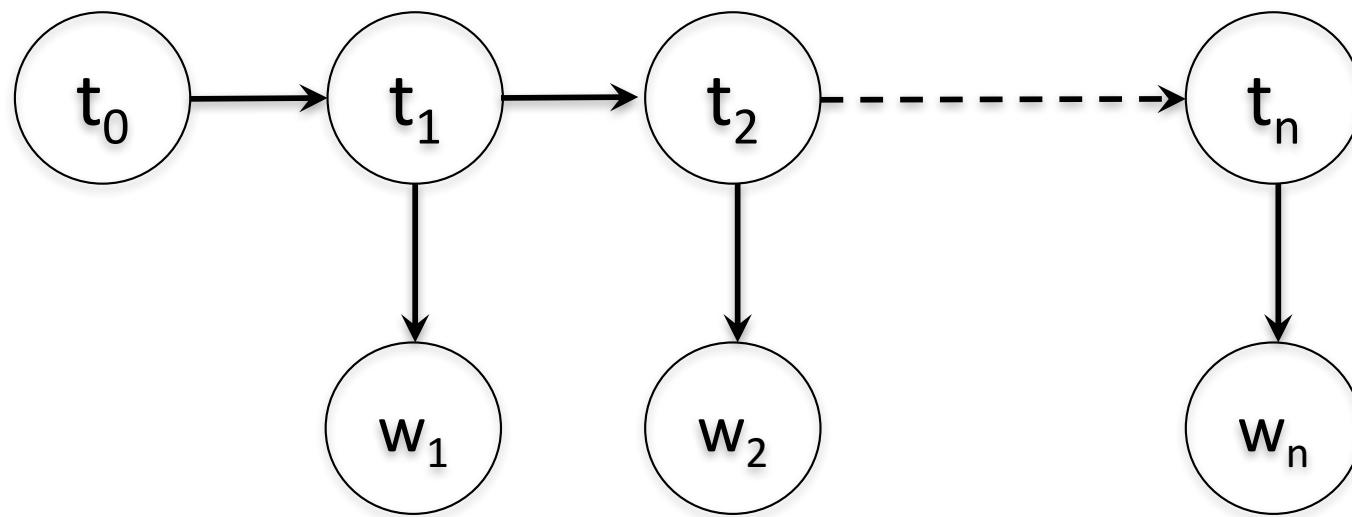
$$p(S|T) = \prod_i p(w_i|t_i) \quad (\text{state emission})$$

- And $p(T)$ into

$$p(T) = p(t_1)p(t_2|t_1)\dots p(t_n|t_{n-1}) \quad (\text{state transition})$$

- Both $p(S|T)$ and $p(T)$ can be estimated using maximum likelihood estimation

POS Tagging with HMMs



$$T_{pred} = \underset{T}{\operatorname{argmax}} P(T|S) \approx \underset{T}{\operatorname{argmax}} \prod_{i=1}^n P(t_i|t_{i-1})P(w_i|t_i)$$

Part-of-Speech Tagging with HMMs, Base Phrase Chunking

UC Davis LIN 127
Spring 2019

Kenji Sagae

POS Tagging with HMMs

- When tagging a word, we look at:
 - The word itself
 - The grammatical environment of the word
- We want the best sequence of tags T for a sentence S

POS Tagging with HMMs

- When tagging a word, we look at:
 - The word itself
 - The grammatical environment of the word
 - We want the best sequence of tags T for a sentence S
 - $\operatorname{argmax}_T p(T|S)$
- $$p(T|S) = \frac{p(S|T)p(T)}{p(S)}$$
- so: $\operatorname{argmax}_T p(T|S) = \operatorname{argmax}_T p(S|T)p(T)$

POS Tagging with HMMs

$$\arg \max_T p(T|S) = \arg \max_T p(S|T)p(T)$$

- We decompose $p(S|T)$ into

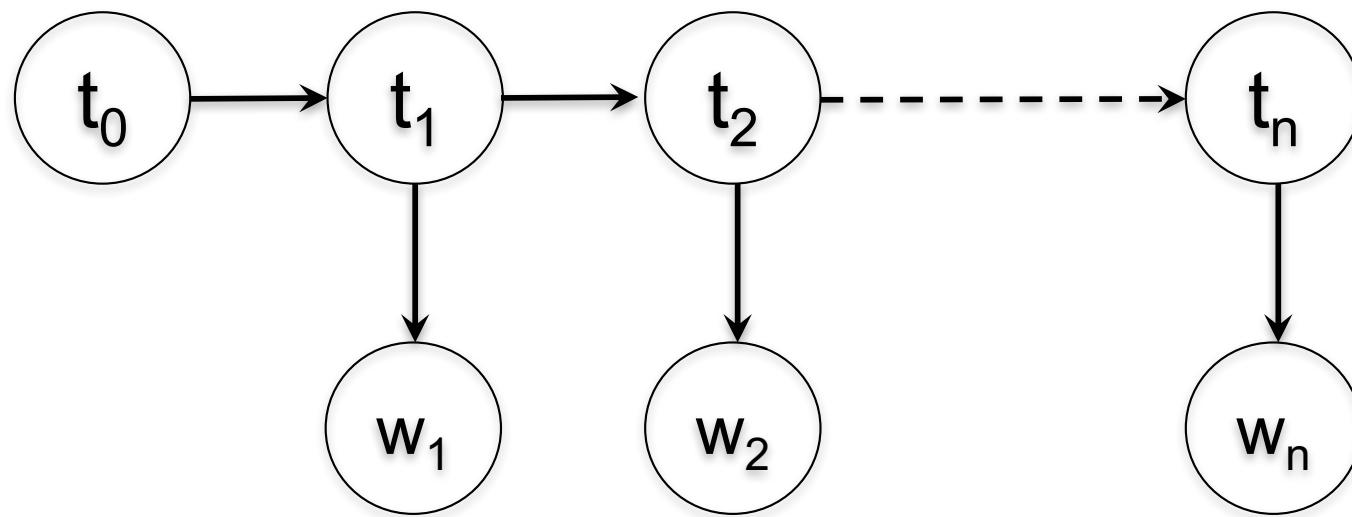
$$p(S|T) = \prod_i p(w_i|t_i) \quad (\text{state emission})$$

- And $p(T)$ into

$$p(T) = p(t_1)p(t_2|t_1)\dots p(t_n|t_{n-1}) \quad (\text{state transition})$$

- Both $p(S|T)$ and $p(T)$ can be estimated using maximum likelihood estimation

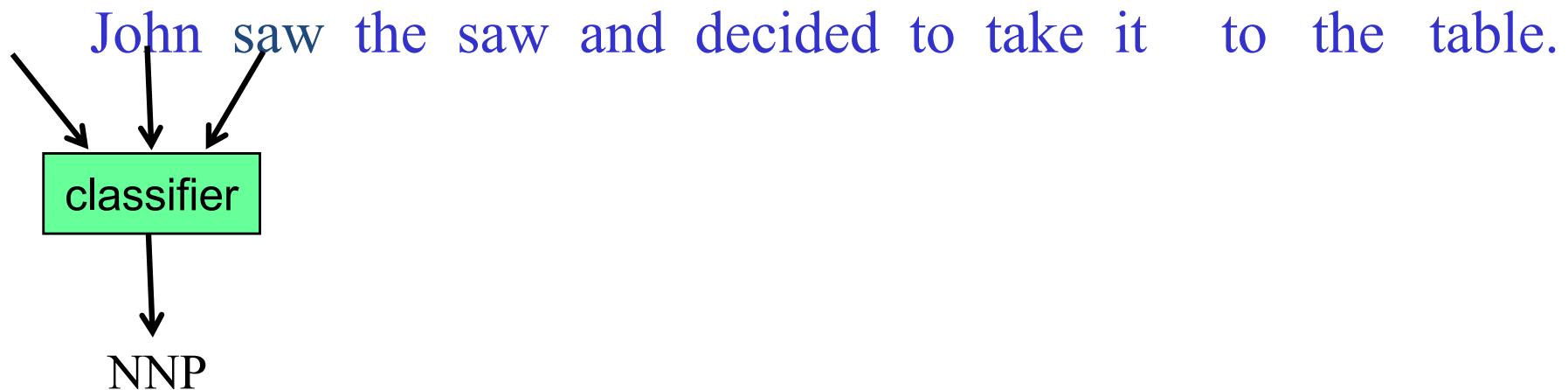
POS Tagging with HMMs



$$T_{pred} = \underset{T}{\operatorname{argmax}} P(T|S) \approx \underset{T}{\operatorname{argmax}} \prod_{i=1}^n P(t_i|t_{i-1})P(w_i|t_i)$$

Sequence Labeling as Classification

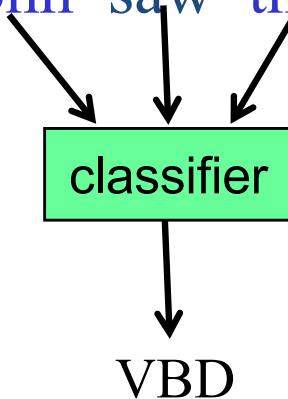
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

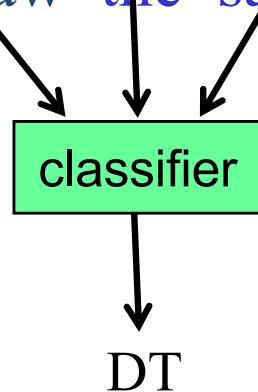
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

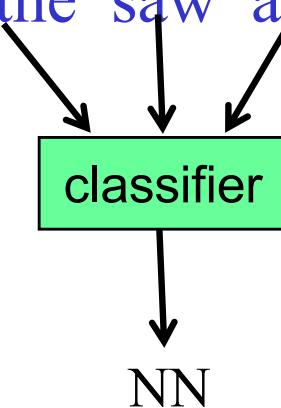
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

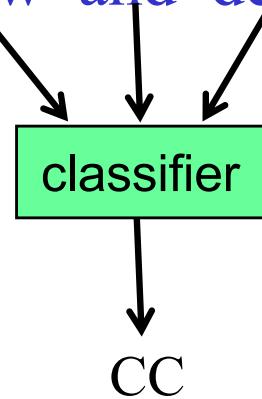
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

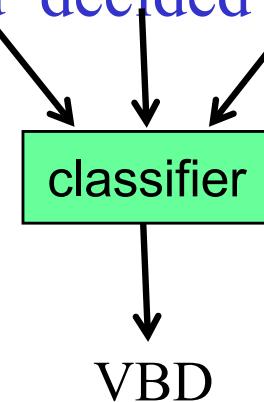
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

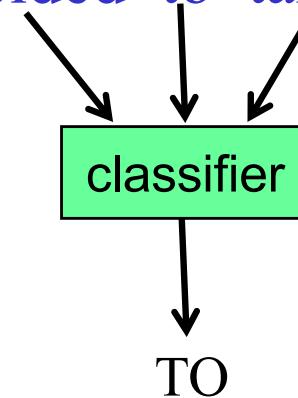
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

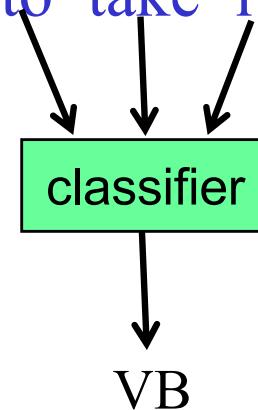
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

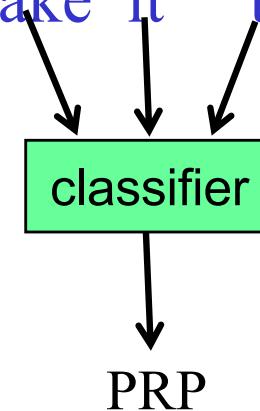
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

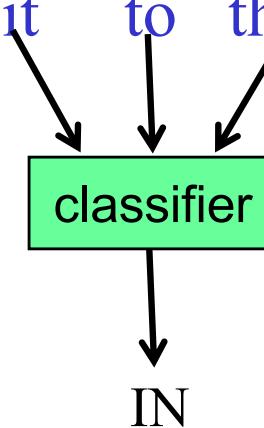
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

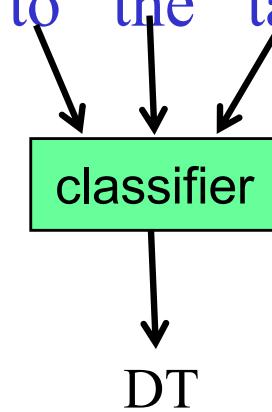
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

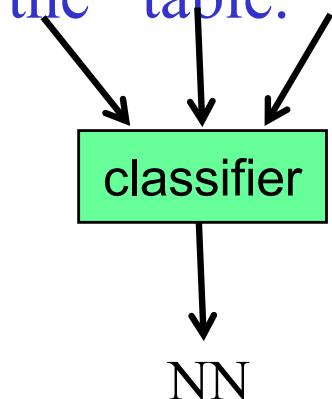
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.



From simple classification to sequence labeling

- Classifying each word individually
 - 93%. Doesn't take context into account
 - (Using spelling features)

From simple classification to sequence labeling

- Classifying each word individually
 - 93%. Doesn't take context into account
 - Features
 - Word, lowercase word
 - Prefix: uncommon => un
 - Suffix: running => ing
 - Is the word capitalized?
 - Word signature
 - All caps, contain numbers, mixed case, special characters, digits, etc.
 - 1970s => Da, 40-year => D-a, McIntosh => AaAa

From simple classification to sequence labeling

- Classifying each word individually
 - 93%. Doesn't take context into account
 - (Using spelling features)
- Using surrounding words as features
 - Better, around 95%.

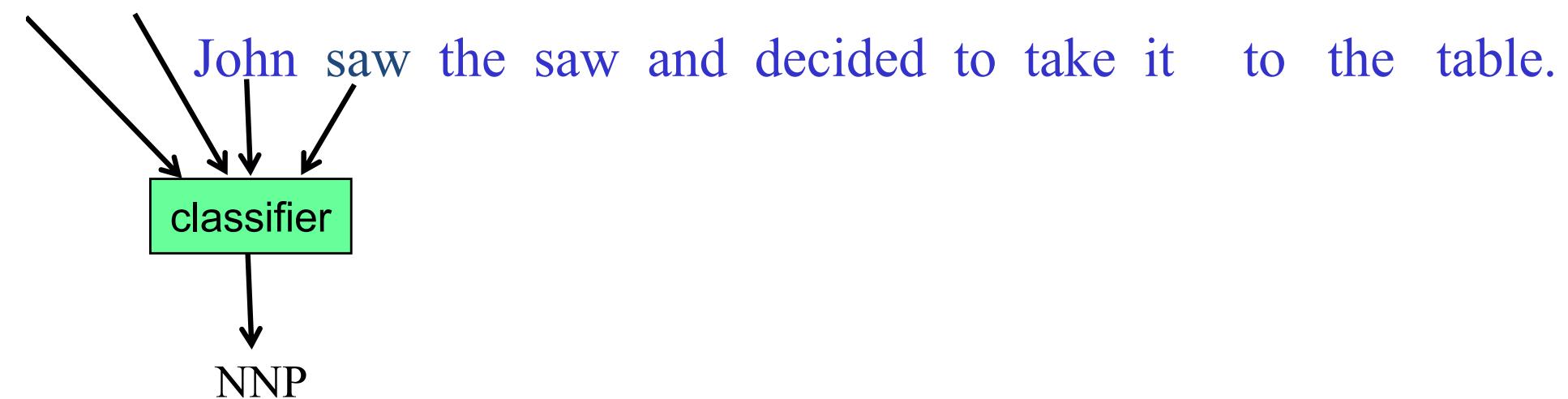
From simple classification to sequence labeling

- Classifying each word individually
 - 93%. Doesn't take context into account
 - (Using spelling features)
- Using surrounding words as features
 - Better, around 95%.
- Using surrounding tags as features
 - Even better, > 96%
 - How?

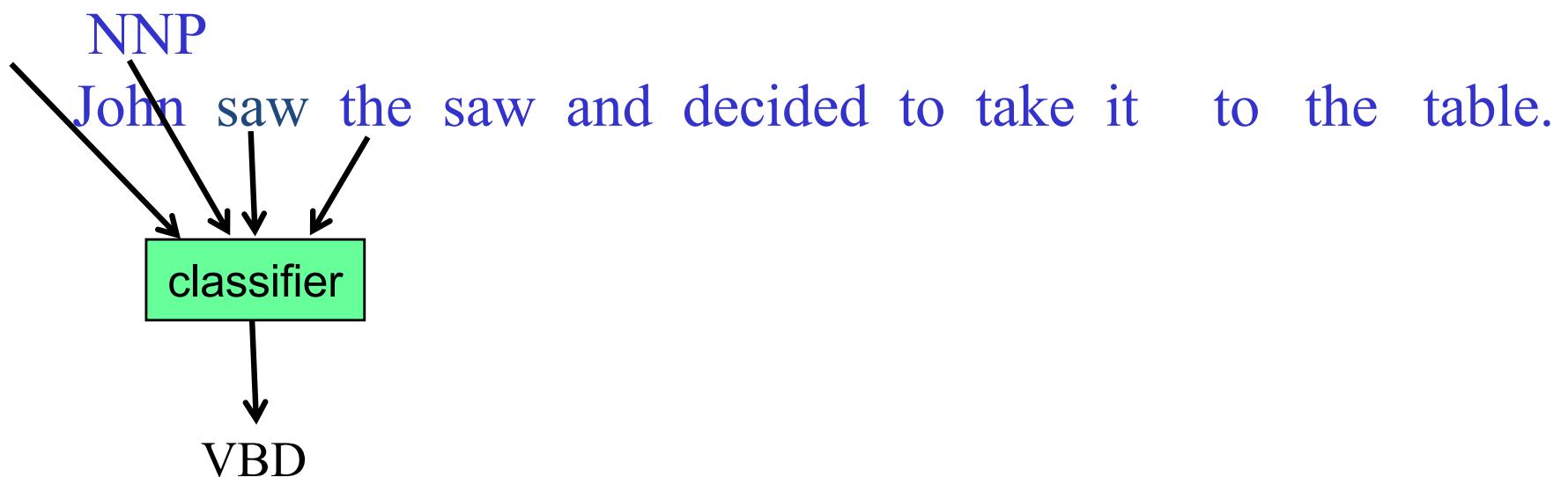
Sequence Labeling as Classification Using Outputs as Inputs

- Better input features are usually the **categories** of the surrounding tokens, but these are not available yet.
- Can use category of either the preceding or succeeding tokens by going forward or back and using previous output.

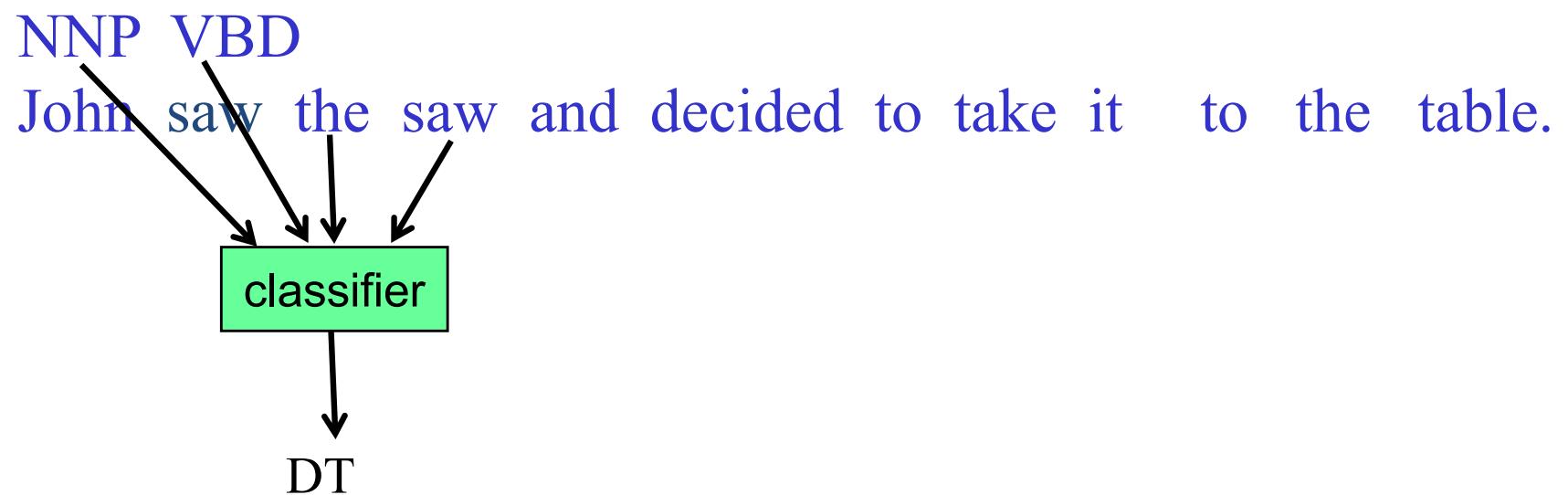
Forward Classification



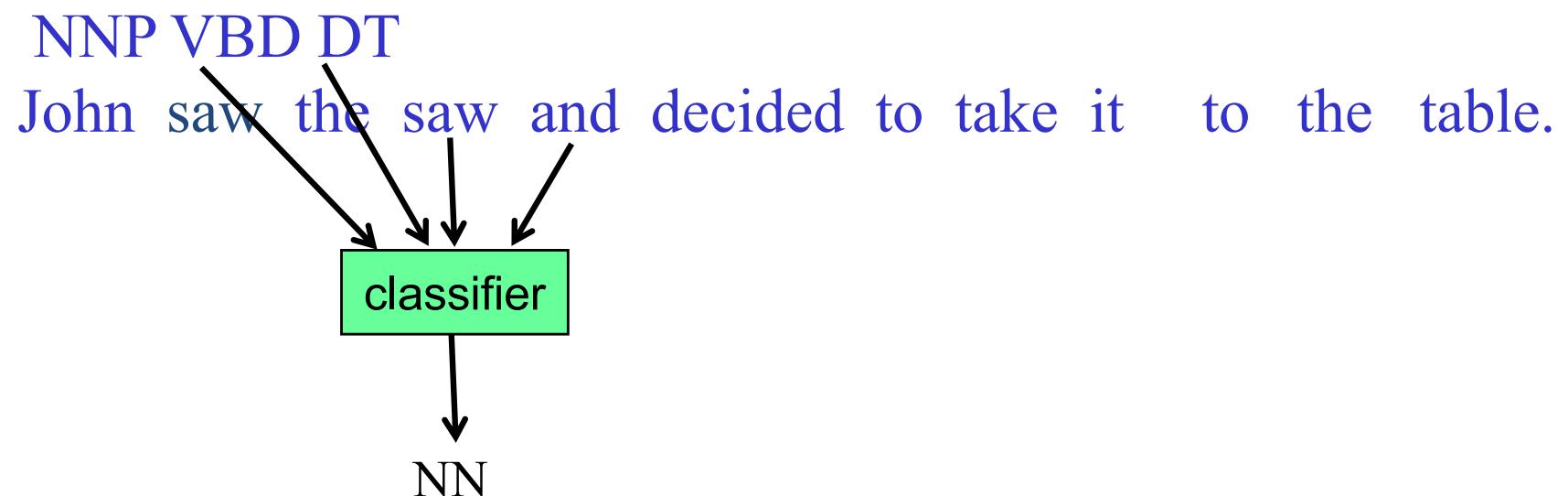
Forward Classification



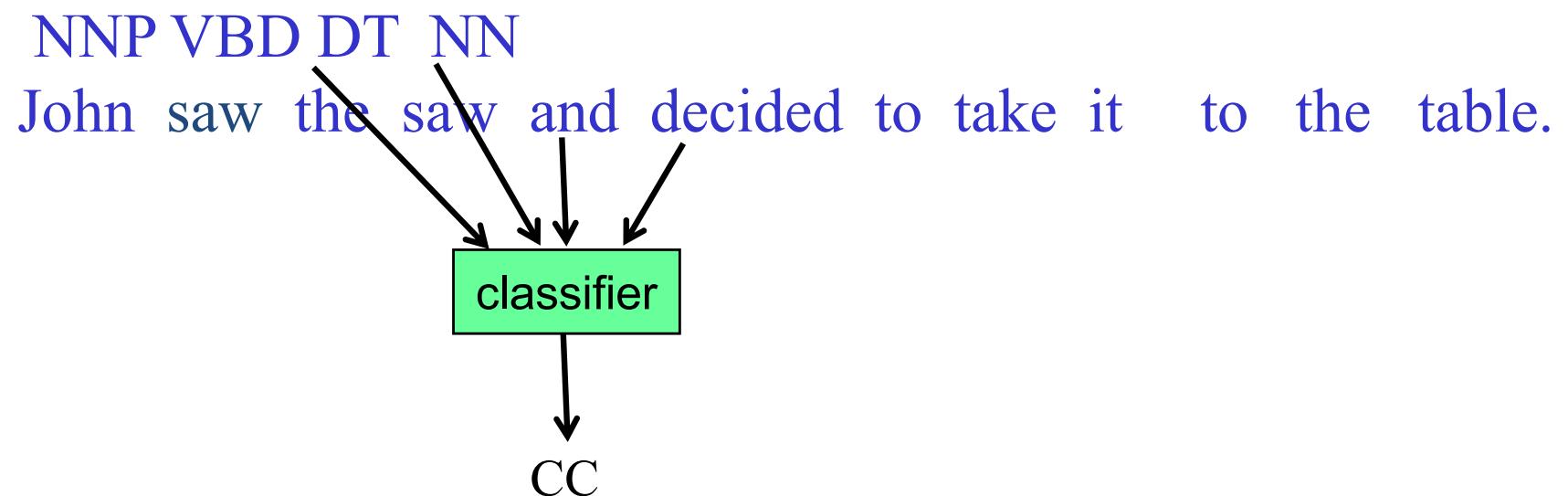
Forward Classification



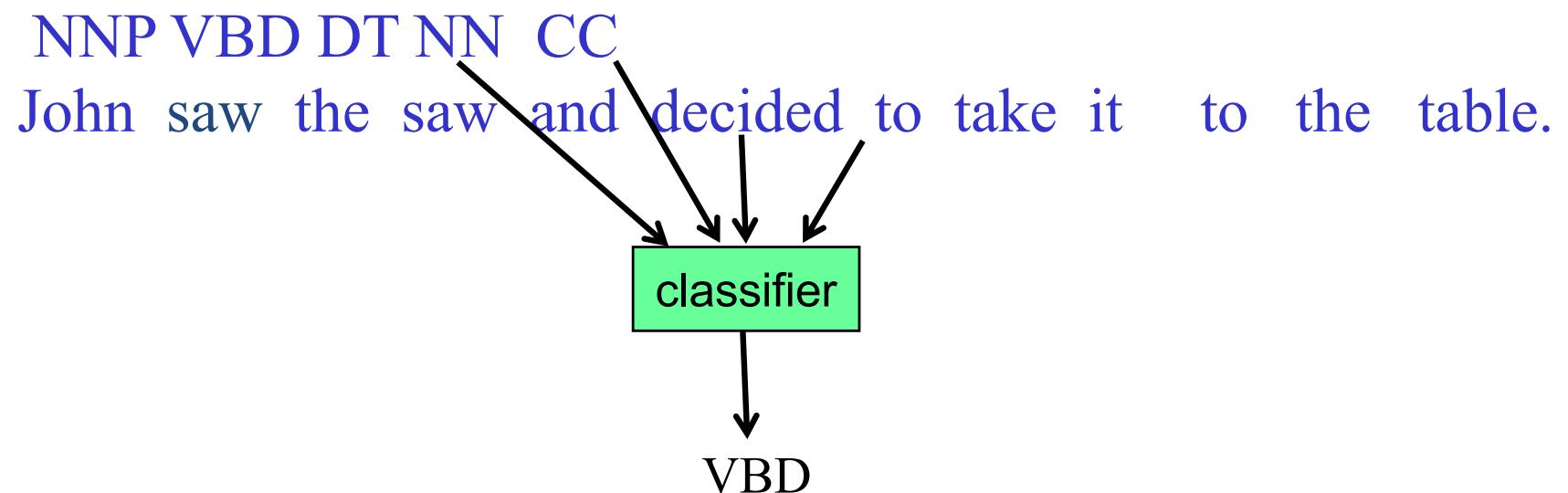
Forward Classification



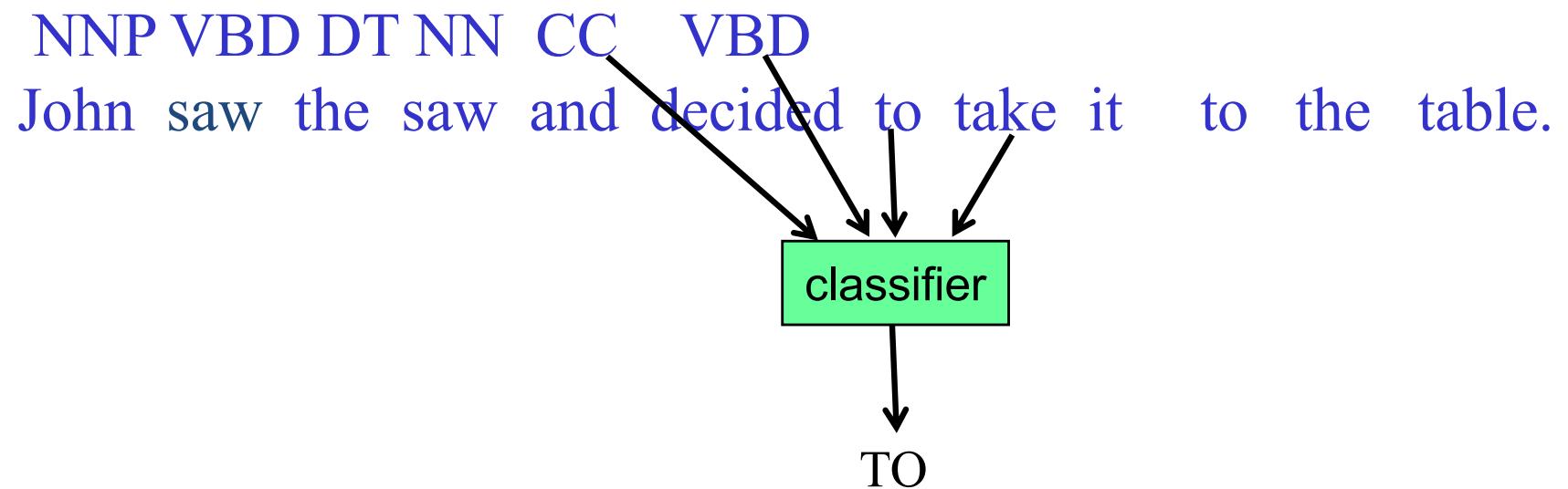
Forward Classification



Forward Classification

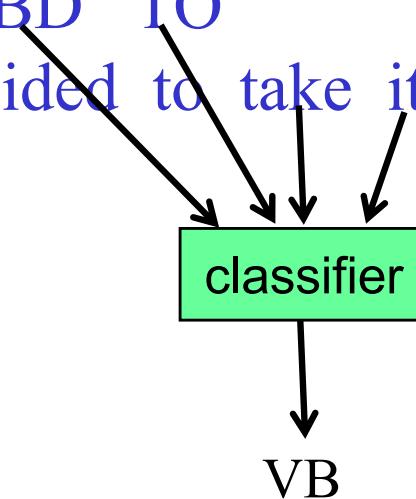


Forward Classification



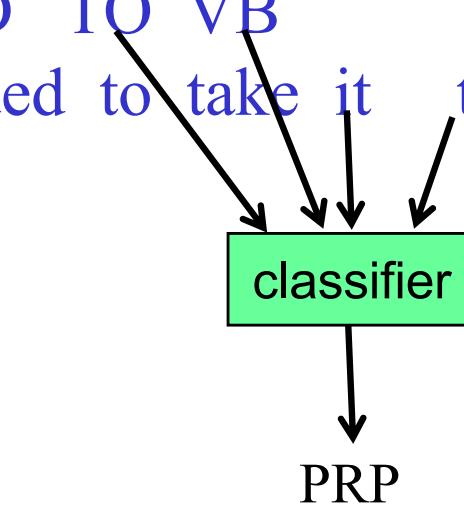
Forward Classification

NNP VBD DT NN CC VBD TO
John saw the saw and decided to take it to the table.



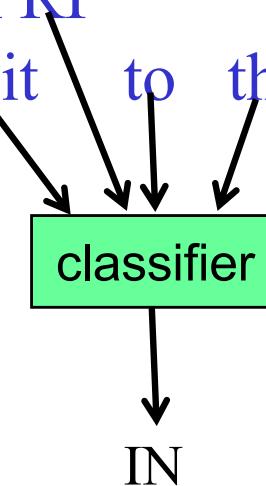
Forward Classification

NNP VBD DT NN CC VBD TO VB
John saw the saw and decided to take it to the table.



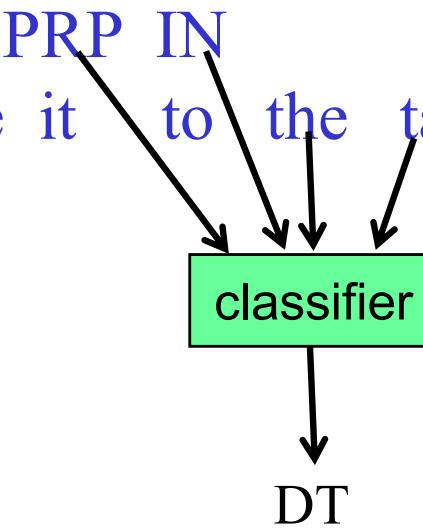
Forward Classification

NNP VBD DT NN CC VBD TO VB PRP
John saw the saw and decided to take it to the table.



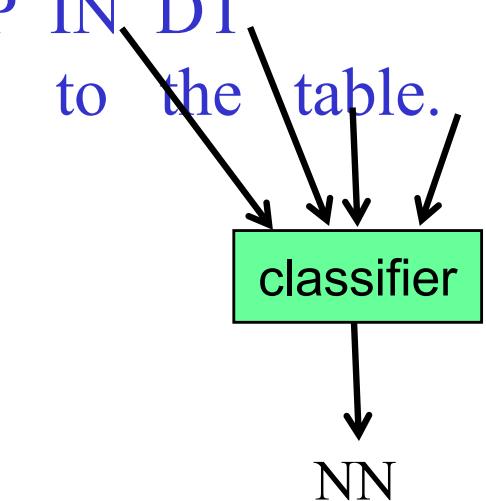
Forward Classification

NNP VBD DT NN CC VBD TO VB PRP IN
John saw the saw and decided to take it to the table.



Forward Classification

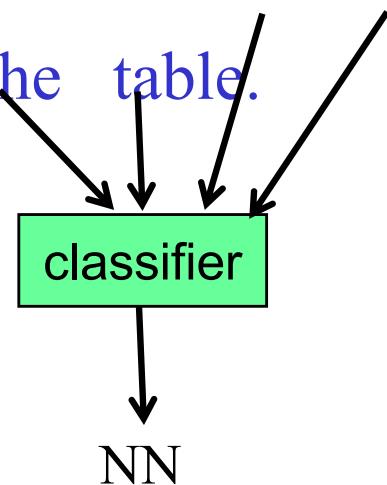
NNP VBD DT NN CC VBD TO VB PRP IN DT
John saw the saw and decided to take it to the table.



Backward Classification

- Disambiguating “to” in this case would be even easier backward.

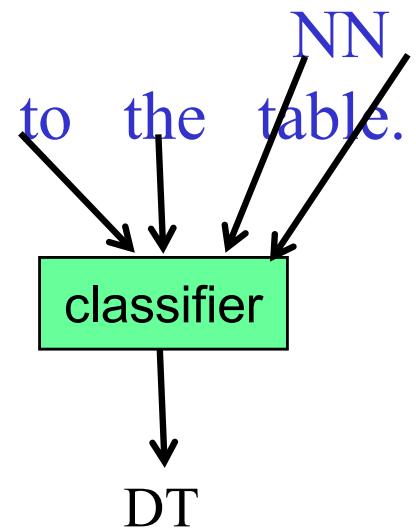
John saw the saw and decided to take it to the table.



Backward Classification

- Disambiguating “to” in this case would be even easier backward.

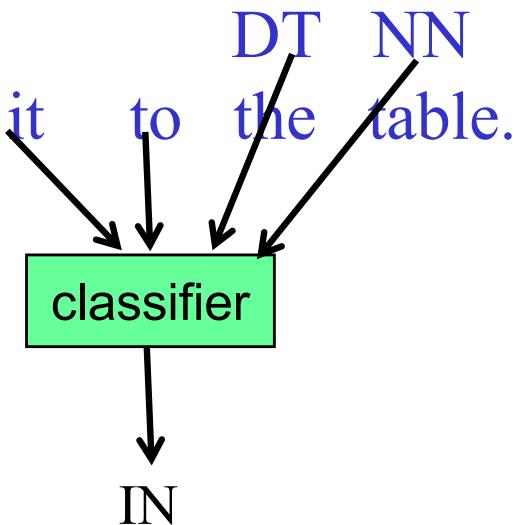
John saw the saw and decided to take it



Backward Classification

- Disambiguating “to” in this case would be even easier backward.

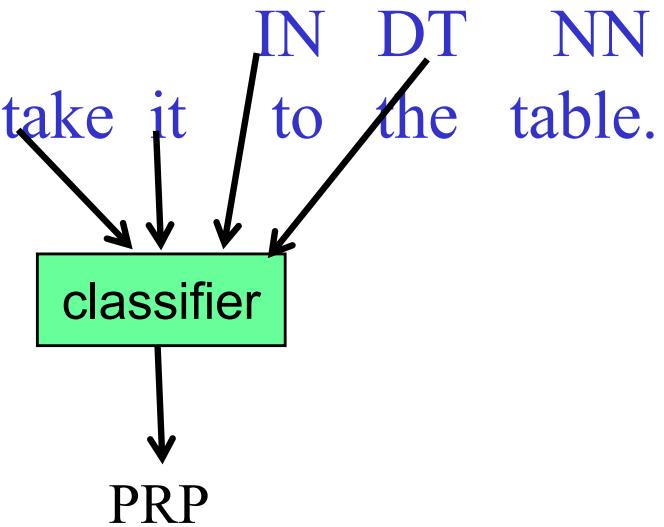
John saw the saw and decided to take it to the table.



Backward Classification

- Disambiguating “to” in this case would be even easier backward.

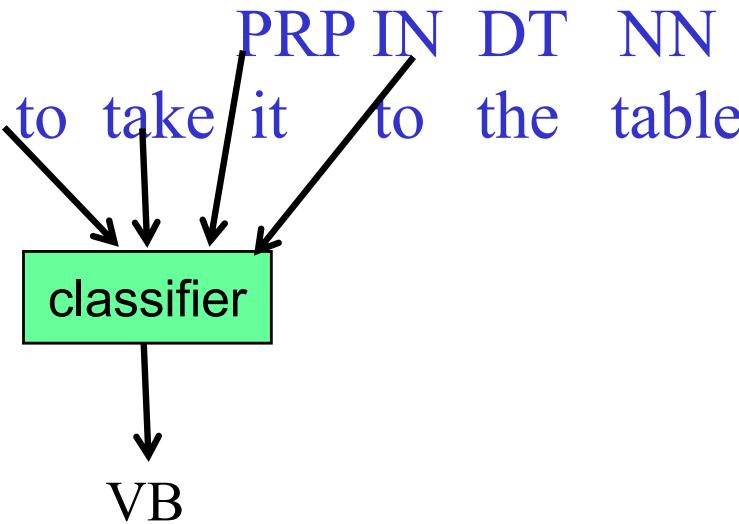
John saw the saw and decided to take it IN to the NN table. DT



Backward Classification

- Disambiguating “to” in this case would be even easier backward.

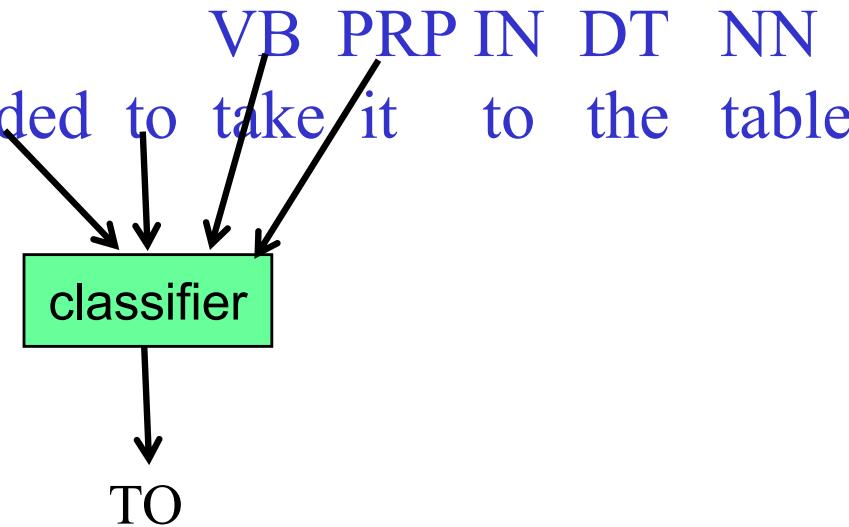
John saw the saw and decided to take it to the table.



Backward Classification

- Disambiguating “to” in this case would be even easier backward.

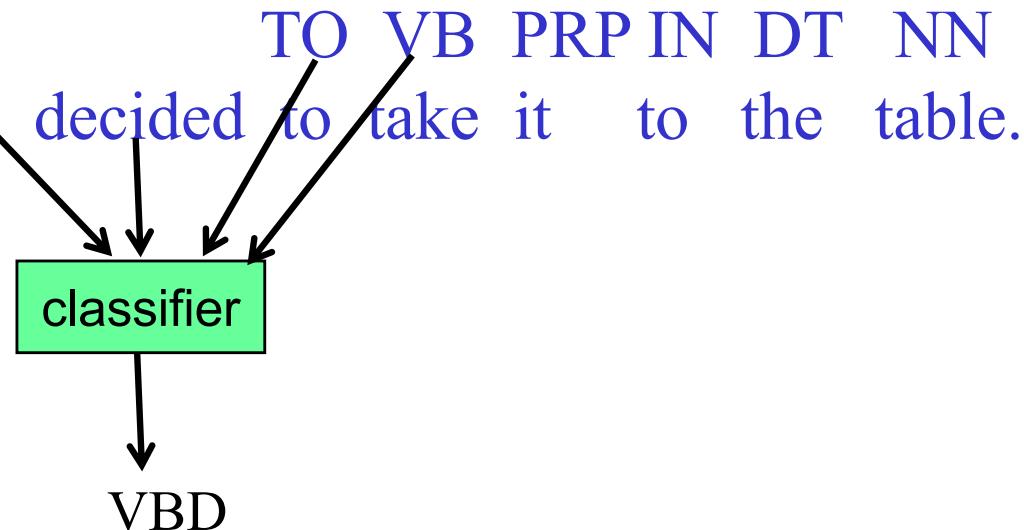
John saw the saw and decided to take it to the table.



Backward Classification

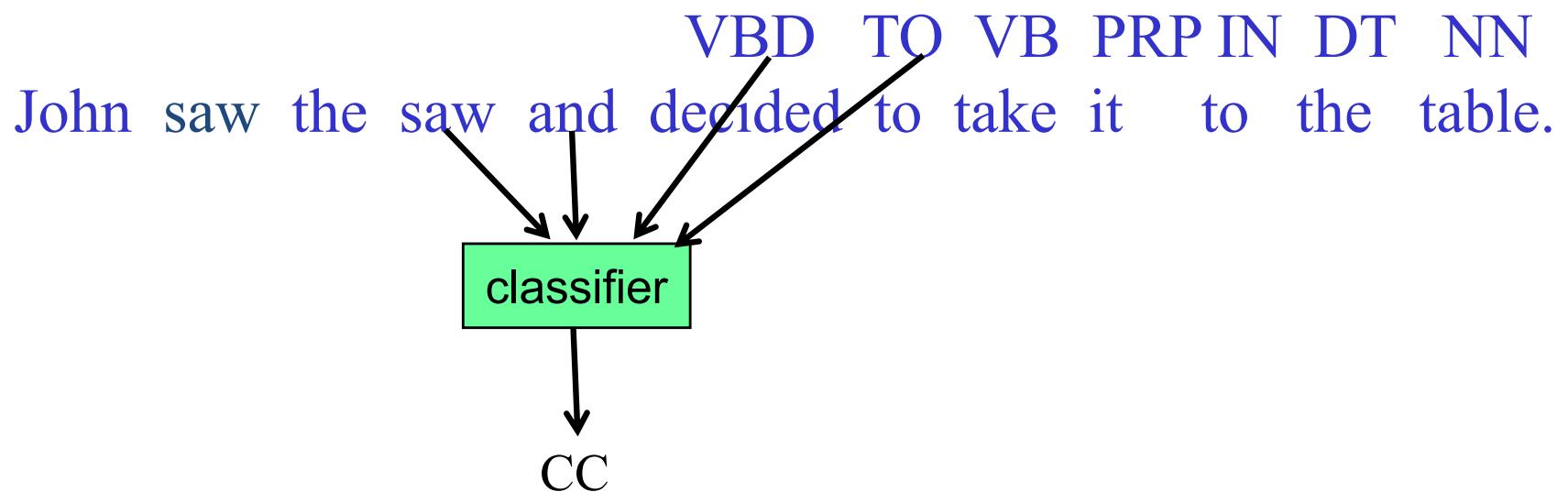
- Disambiguating “to” in this case would be even easier backward.

John saw the saw and decided to take it to the table.
TO VB PRP IN DT NN



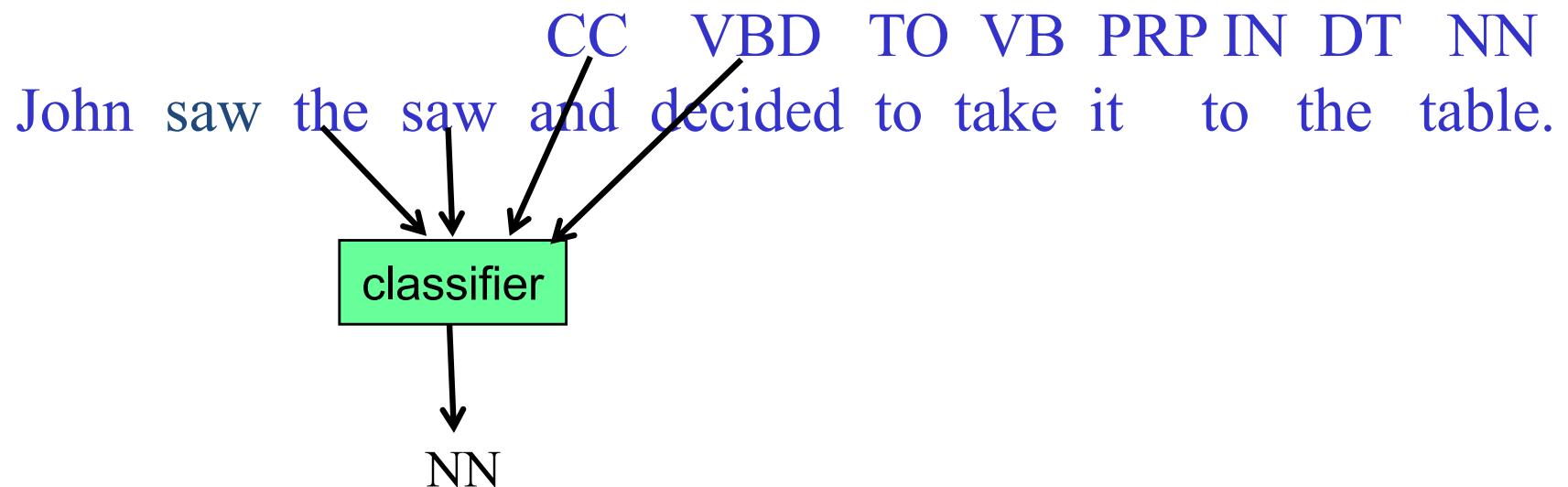
Backward Classification

- Disambiguating “to” in this case would be even easier backward.



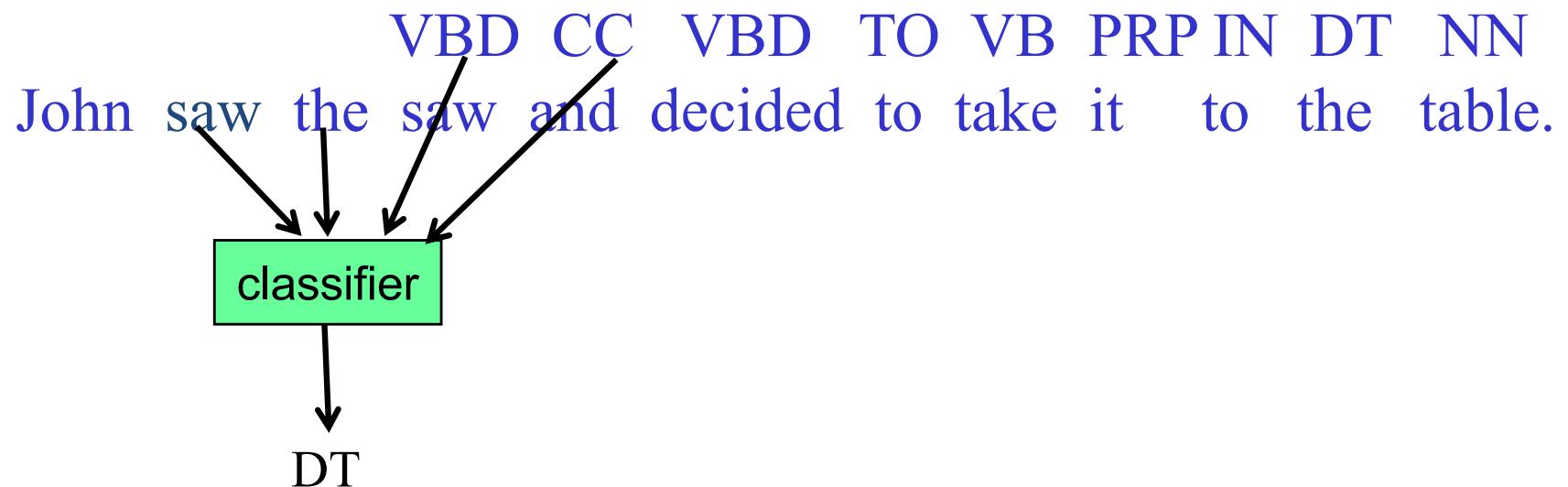
Backward Classification

- Disambiguating “to” in this case would be even easier backward.



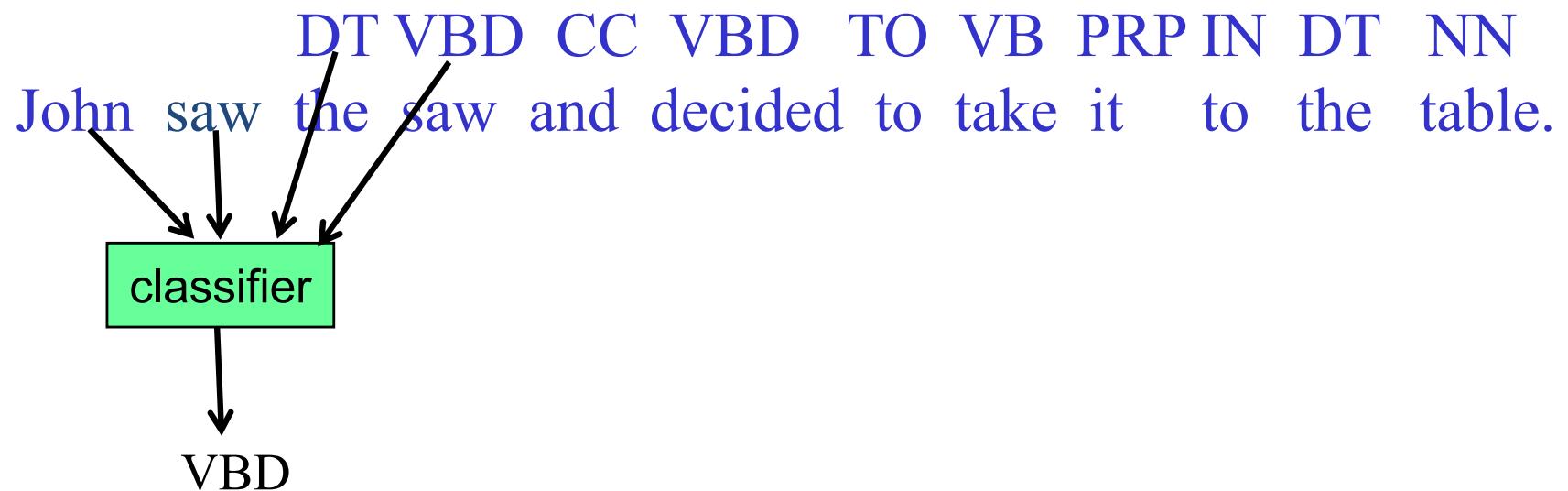
Backward Classification

- Disambiguating “to” in this case would be even easier backward.



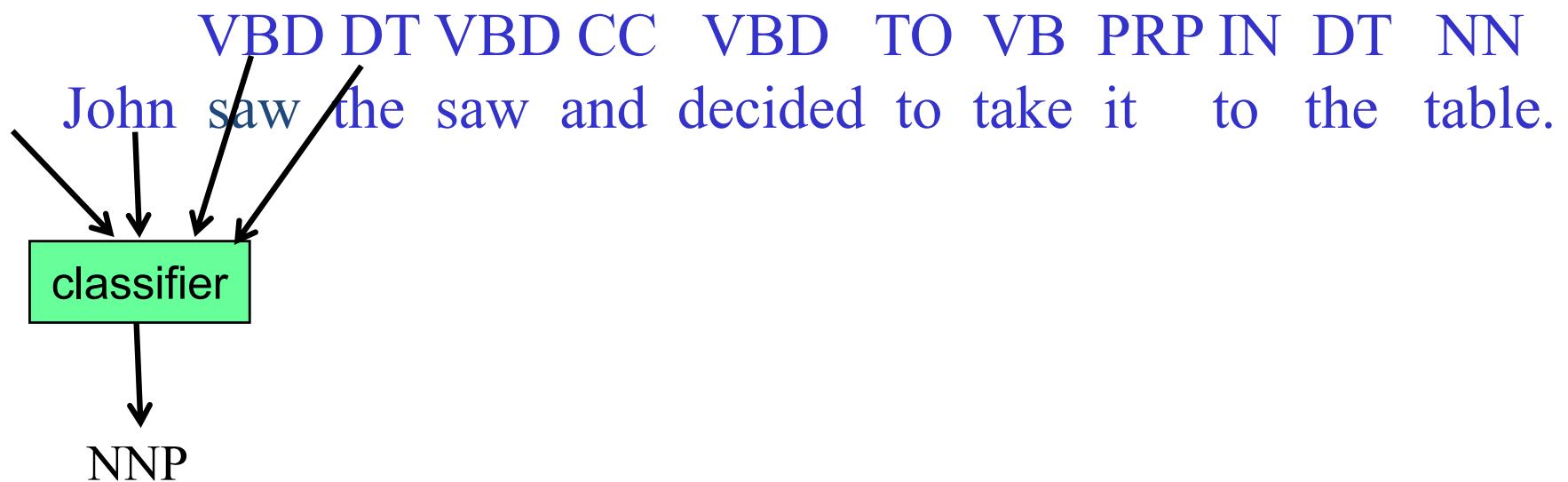
Backward Classification

- Disambiguating “to” in this case would be even easier backward.



Backward Classification

- Disambiguating “to” in this case would be even easier backward.



Problems with Sequence Labeling as Classification

- Not easy to integrate information from category of tokens on both sides.
- Difficult to propagate uncertainty between decisions and “collectively” determine the most likely joint assignment of categories to all of the tokens in a sequence.
- Still high POS tagging accuracy

Probabilistic Sequence Models

- Probabilistic sequence models allow integrating uncertainty over multiple, interdependent classifications and collectively determine the most likely global assignment.
- Two standard models
 - Hidden Markov Model (HMM)
 - Conditional Random Field (CRF)
 - Used in many sequence labeling problems in NLP
 - But don't help that much for POS tagging accuracy

Are POS taggers really 97% accurate?

- Yes, if you want to tag Wall Street Journal text
 - Especially WSJ from the late 80s
 - Taggers typically perform very well on text of the same genre and domain as training data
 - The most popular training set is the Penn Treebank
 - WSJ text
 - Also Brown corpus and Switchboard
- Severe degradation in accuracy in other domains and other genres of language
 - Biomedical text, blogs, novels, etc.
- How to get high accuracy in other genres?
 - Annotate domain-specific data
 - Domain adaptation

What else can we tag?

- Shallow parsing
 - Base phrases
 - Chunking
 - [NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only \$ 1.8 billion] [PP in] [NP September] .
- Named entity recognition
- And much more

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP

He reckons the current deficit will narrow to only 1.8 billion

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP VP

He reckons the current deficit will narrow to only 1.8 billion

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP VP NP
He reckons the current deficit will narrow to only 1.8 billion

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP VP NP NP
He reckons the current deficit will narrow to only 1.8 billion

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP VP NP NP NP
He reckons the current deficit will narrow to only 1.8 billion

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP VP NP NP NP VP VP
He reckons the current deficit will narrow to only 1.8 billion

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP VP NP NP VP VP TO NP NP NP
He reckons the current deficit will narrow to only 1.8 billion

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP
He

VP
reckons

NP NP NP
the current deficit

VP VP
will narrow

TO
to

NP NP NP
only 1.8 billion

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP
He

VP
reckons

NP NP NP
the current deficit

VP VP
will narrow

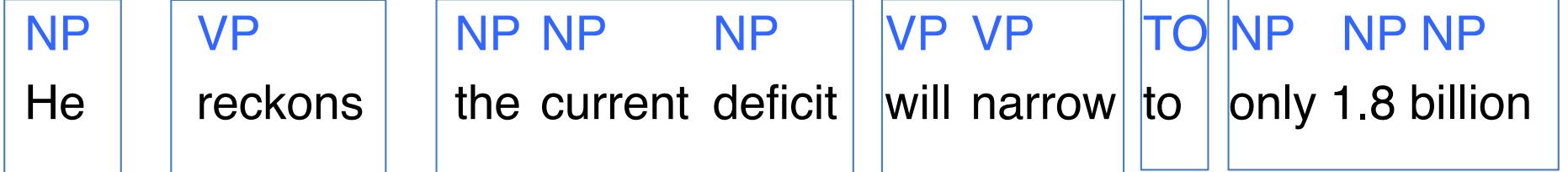
TO
to

NP NP NP
only 1.8 billion

What is the problem here?

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

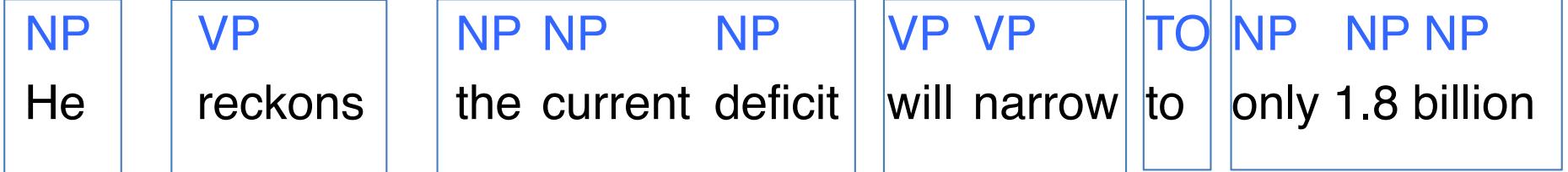


What is the problem here?

I gave the student a book

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

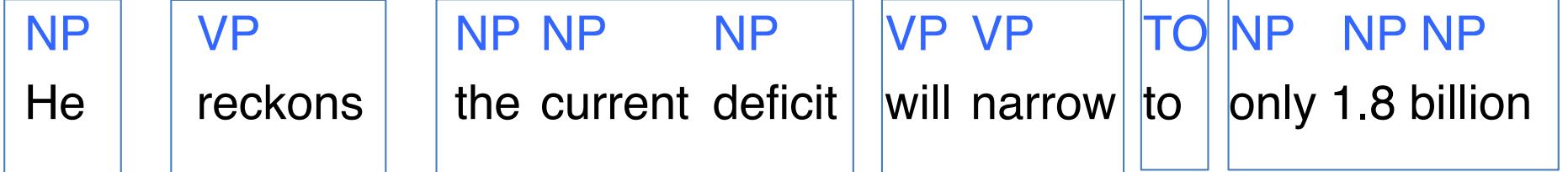


What is the problem here?

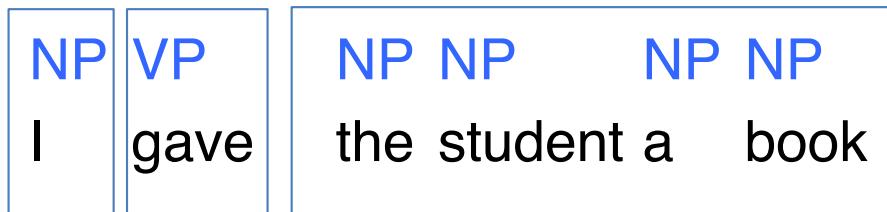
NP VP NP NP NP NP
I gave the student a book

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

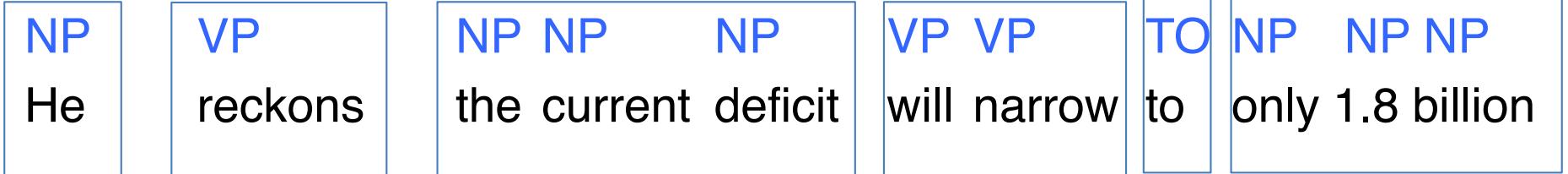


What is the problem here?

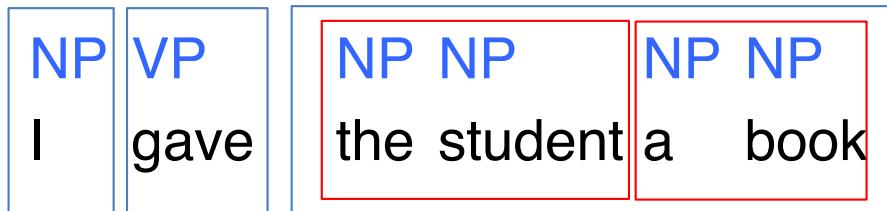


Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]



What is the problem here?



Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

- Words beginning chunks are marked as B
- Words continuing chunks are marked as I
- Words not belonging to a chunk are marked as O

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP B-NP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP B-NP I-NP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP B-NP I-NP I-NP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP B-NP I-NP I-NP O
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP	B-VP	B-NP I-NP I-NP	B-VP I-VP	B-PP	B-NP I-NP I-NP	O
He	reckons	the current deficit	will narrow	to	only 1.8 billion	.

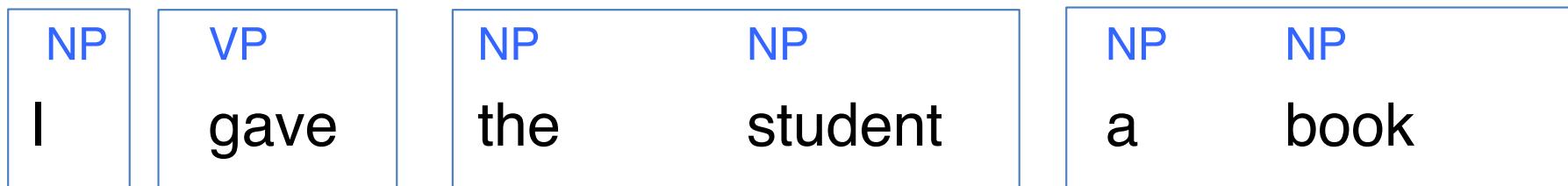
Shallow Parsing

I gave the student a book

Shallow Parsing

NP VP NP NP NP NP
I gave the student a book

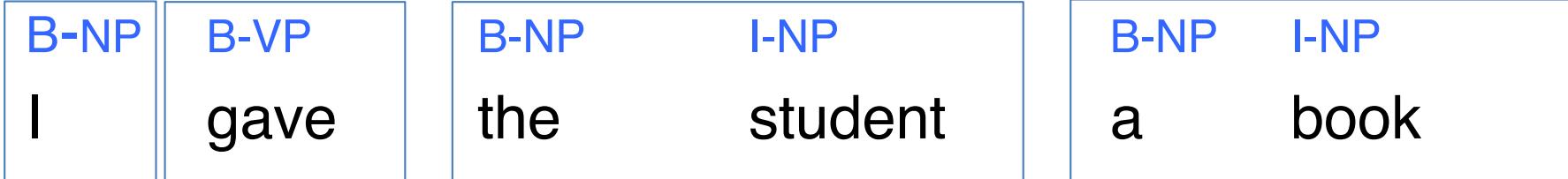
Shallow Parsing



Shallow Parsing

I gave the student a book

Shallow Parsing



Named Entity Recognition, Word Classes and Relations

UC Davis LIN 127
Spring 2019

Kenji Sagae

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

NP
He

VP
reckons

NP NP NP
the current deficit

VP VP
will narrow

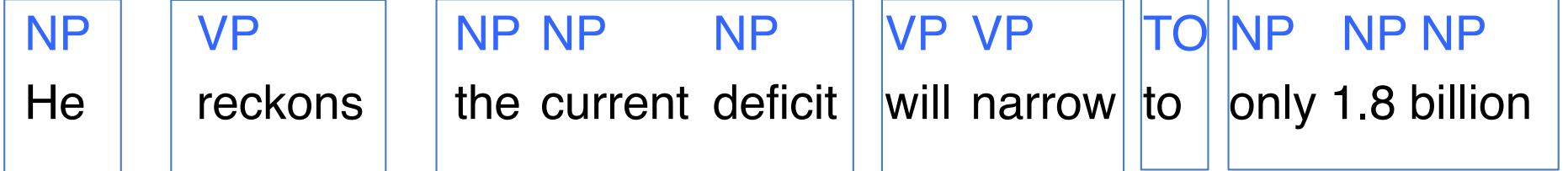
TO
to

NP NP NP
only 1.8 billion

What is the problem here?

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

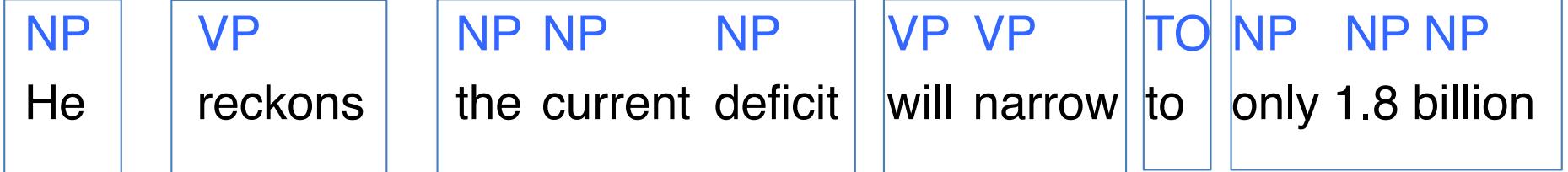


What is the problem here?

I gave the student a book

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

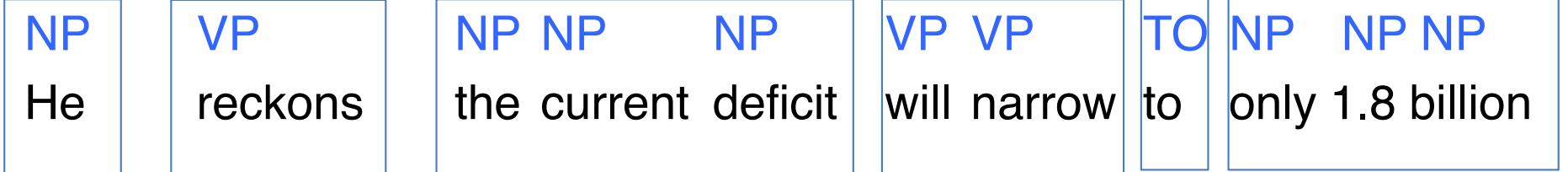


What is the problem here?

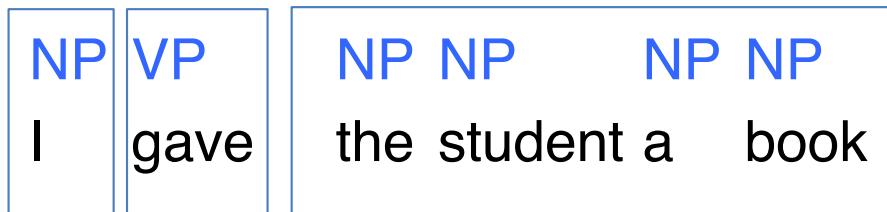
NP VP NP NP NP NP
I gave the student a book

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

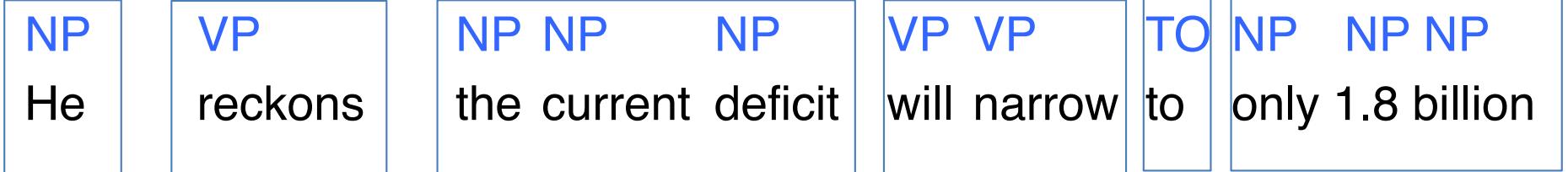


What is the problem here?

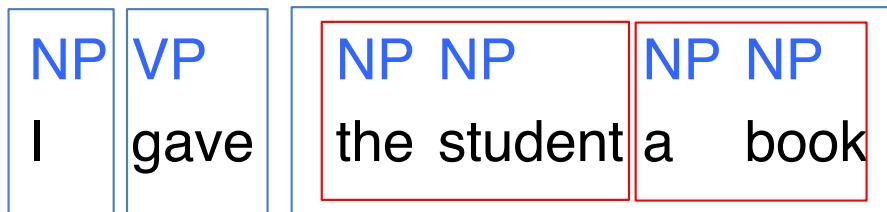


Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]



What is the problem here?



Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

- Words beginning chunks are marked as B
- Words continuing chunks are marked as I
- Words not belonging to a chunk are marked as O

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP

He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP B-NP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP B-NP I-NP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP B-NP I-NP I-NP
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP B-VP B-NP I-NP I-NP B-VP I-VP B-PP B-NP I-NP I-NP O
He reckons the current deficit will narrow to only 1.8 billion .

Shallow Parsing

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow]
[PP to] [NP only \$ 1.8 billion]

BIO Encoding

B-NP	B-VP	B-NP I-NP I-NP	B-VP I-VP	B-PP	B-NP I-NP I-NP	O
He	reckons	the current deficit	will narrow	to	only 1.8 billion	.

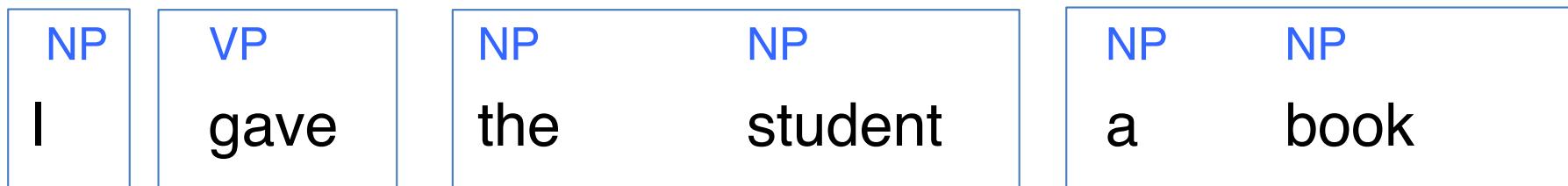
Shallow Parsing

I gave the student a book

Shallow Parsing

NP VP NP NP NP NP
I gave the student a book

Shallow Parsing



Shallow Parsing

I gave the student a book

Shallow Parsing

B-NP

I

B-VP

gave

B-NP

the

I-NP

student

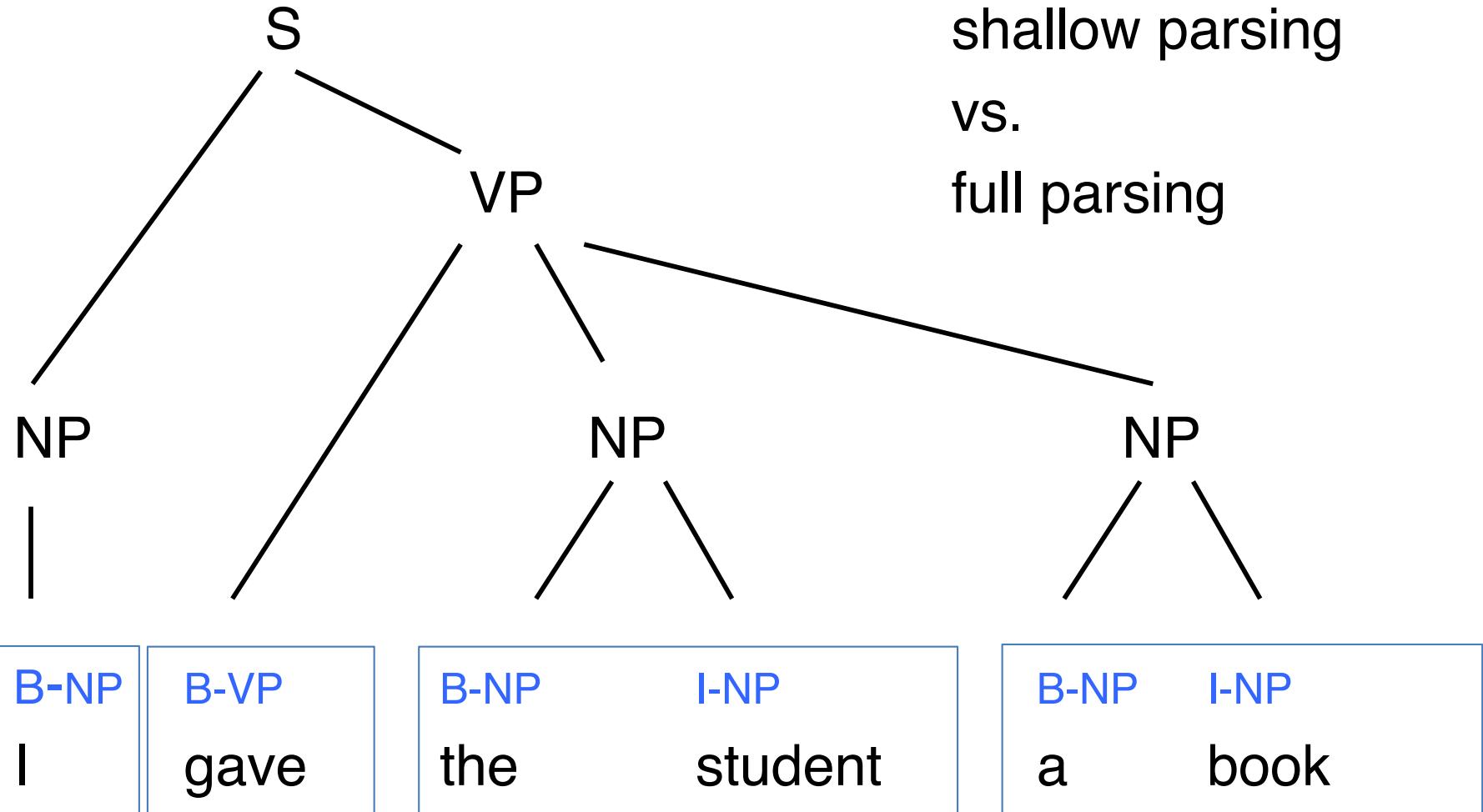
B-NP

a

I-NP

book

Shallow Parsing



Named Entity Recognition

Adam Smith works for **IBM**, **London** since **February 2010**.

- Identify mentions in text and classify them into a predefined set of categories of interest:
 - Person: **Adam Smith**
 - Organizations: **IBM**
 - Locations: **London**
 - Date: **February 2010**

NER as sequence labeling

- Original sentence
 - Adam Smith works for IBM, London since Feb 2011
- Tokenized
 - Adam Smith works for IBM , London since Feb 2011
- Tags (one per token)

Adam Smith works for IBM , London since Feb 2011

NER as sequence labeling

- Original sentence
 - Adam Smith works for IBM, London since Feb 2010
- Tokenized
 - Adam Smith works for IBM , London since Feb 2011
- Tags (one per token)

PER

Adam Smith

ORG

IBM

LOC

London

DATE

Feb 2011

NER as sequence labeling

- Original sentence
 - Adam Smith works for IBM, London since Feb 2010
- Tokenized
 - Adam Smith works for IBM , London since Feb 2011
- Tags (one per token)

B-PER I-PER

Adam Smith

B-ORG

works for IBM,

B-LOC

London

B-DATE I-DATE

since Feb 2011

NER as sequence labeling

- Original sentence
 - Adam Smith works for IBM, London since Feb 2010
- Tokenized
 - Adam Smith works for IBM , London since Feb 2011
- Tags (one per token)

B-PER	I-PER	O	O	B-ORG	O	B-LOC	O	B-DATE	I-DATE		
Adam	Smith		works	for	IBM	,	London		since	Feb	2011

NER as sequence labeling

- Original sentence
 - Adam Smith works for IBM, London since Feb 2010
- Tokenized
 - Adam Smith works for IBM , London since Feb 2011
- Tags (one per token)

B-PER I-PER O O B-ORG O B-LOC O B-DATE I-DATE
Adam Smith works for IBM , London since Feb 2011

Word Classes and Relations

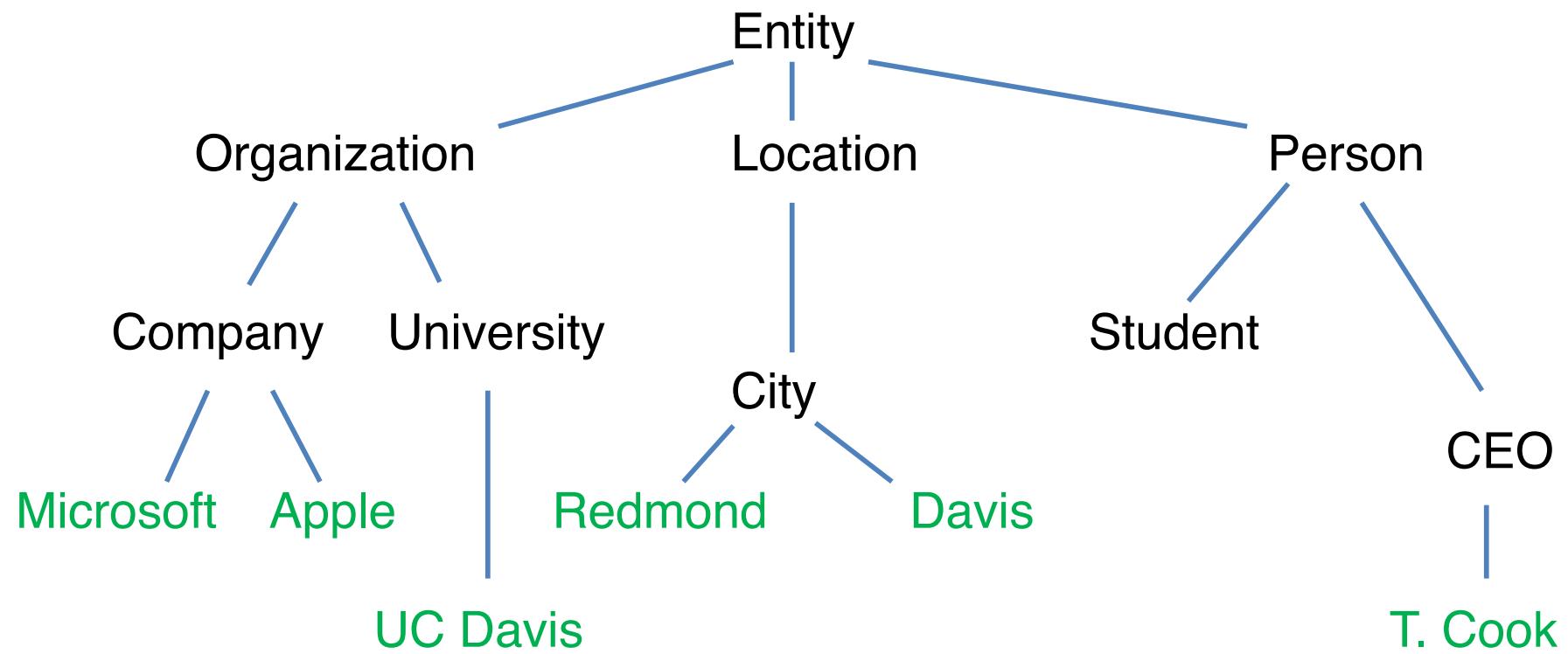
Semantic Classes

- A semantic class contains words that share a semantic property
- Examples
 - Nobel prize winners: Albert Einstein, Max Plank, ...
 - European Union countries: France, Germany, ...
 - People: man, woman, boy, girl, student, Bob, ...

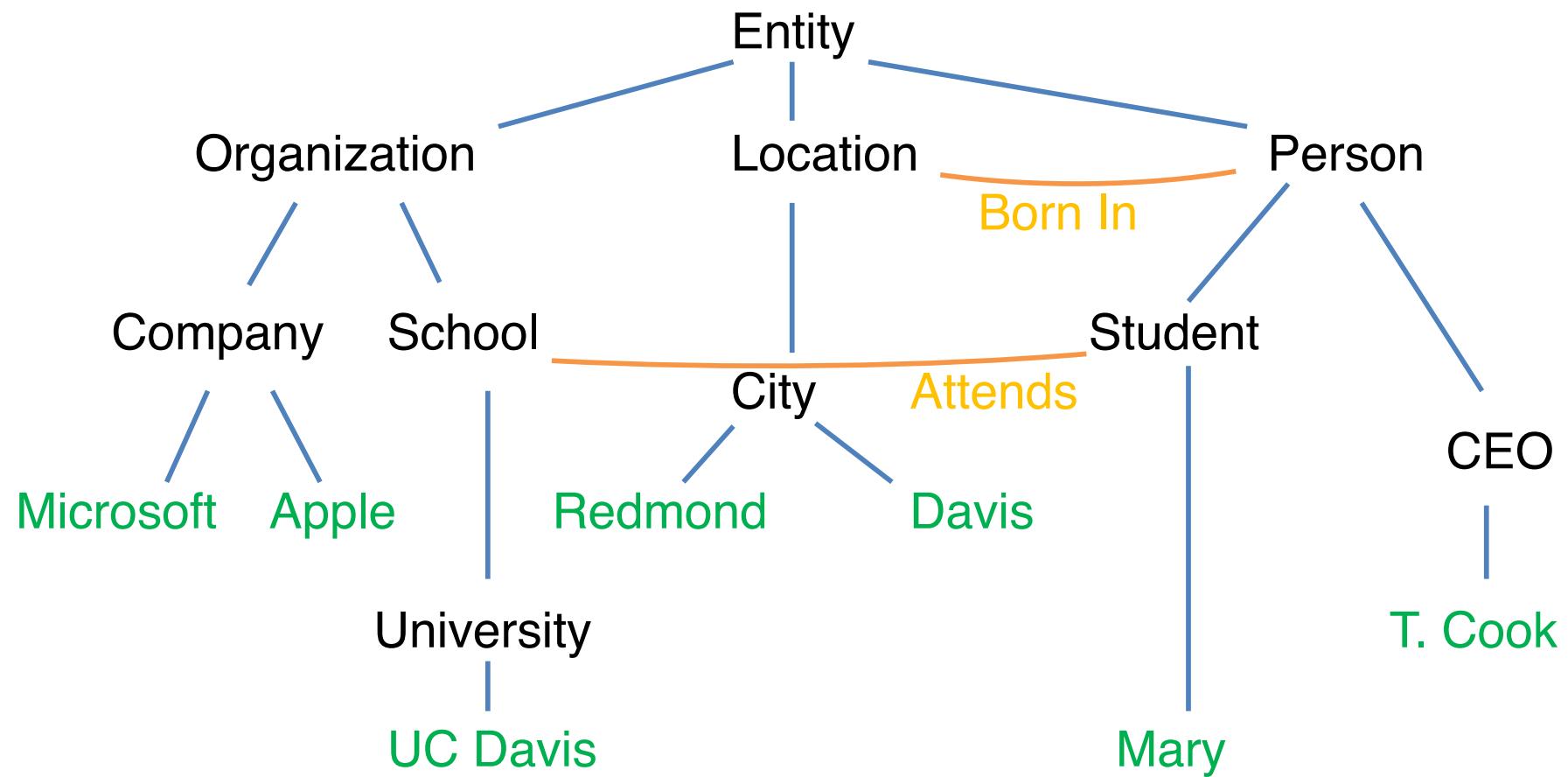
Characteristics of Semantic Classes

- Based on instance membership
 - Closed
 - Small: names of countries, planets
 - Large: names of diseases, cities
 - Open
 - Movies, books, singers, etc.
- An instance can belong to multiple classes
 - Michael Jordan
 - Basketball player
 - Baseball player
 - Computer Scientist? Actor?
 - No, homonymy
- Instances and classes are bound by diverse relations

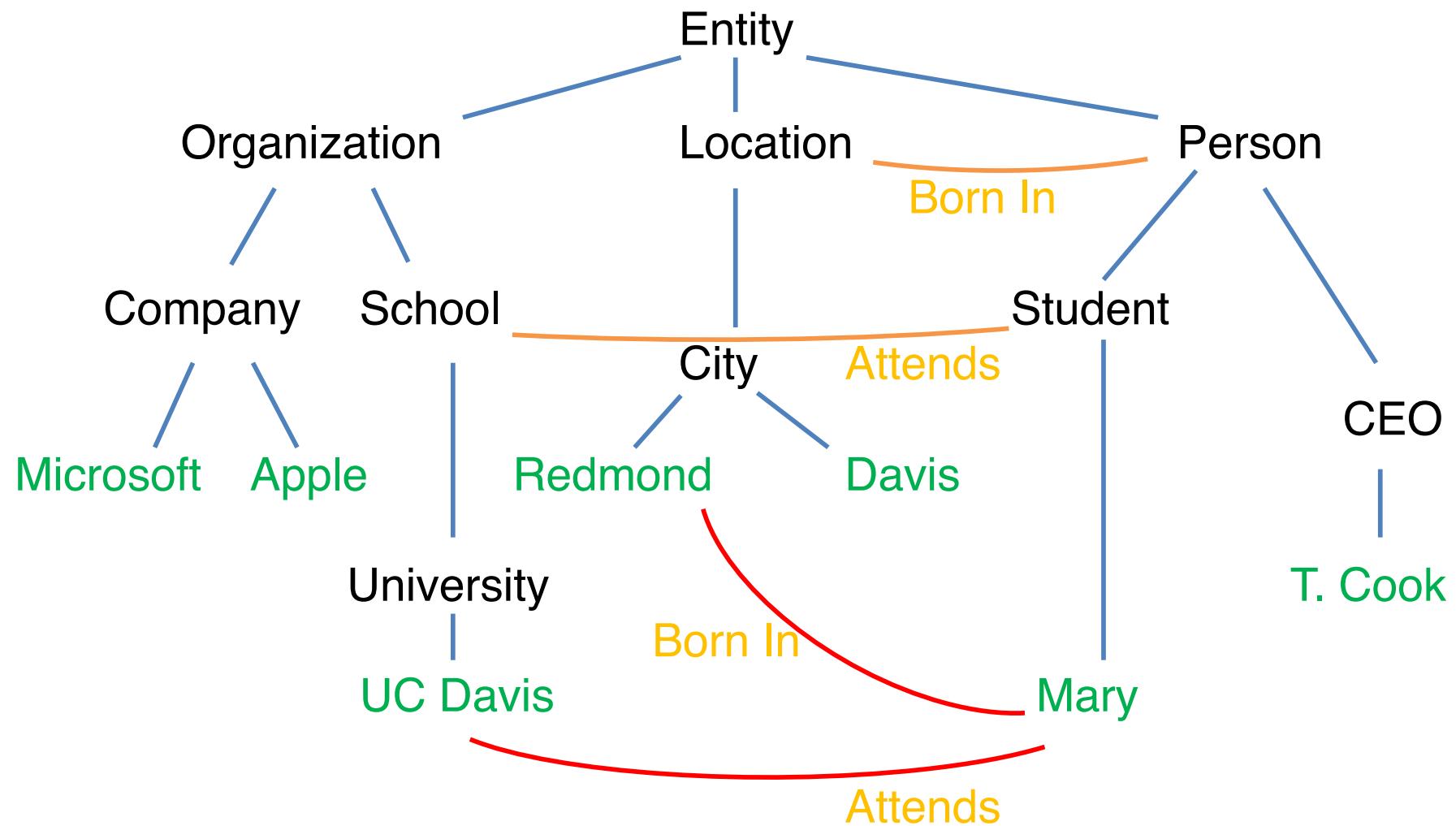
Relations



Relations

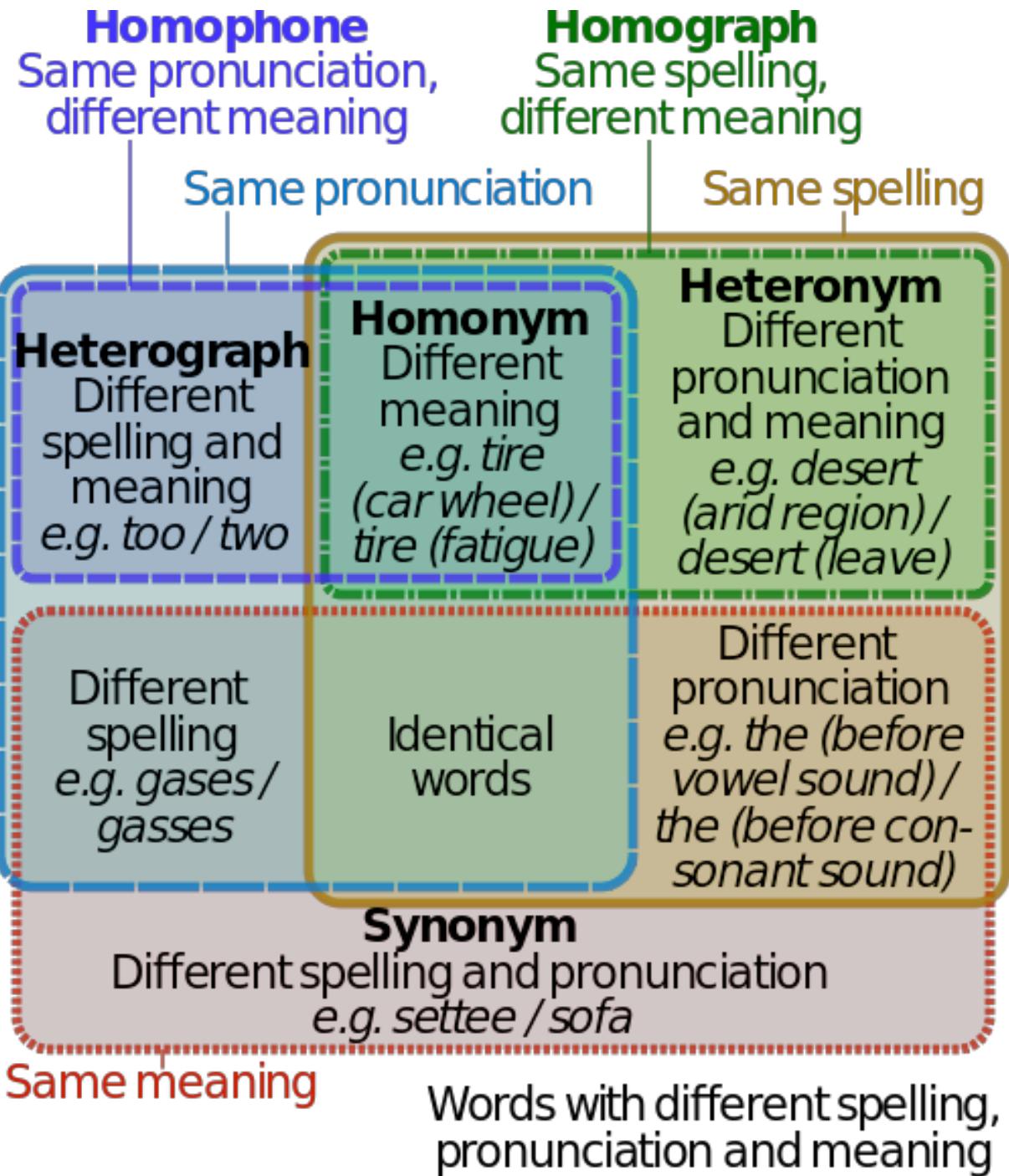


Relations



Homonymy

- “Same name”
- Words that have the same spelling and pronunciation, but distinct unrelated meanings
 - Example: bat (animal) and bat (baseball equipment)
 - Some definitions require spelling or pronunciation
- Homographs
 - Same spelling
 - Example: bass (instrument) and bass (fish)
- Homophones
 - Same sound
 - Example: piece and peace



Source: Wikimedia commons

Polysemy

- Polysemous words have multiple (related) senses
 - Example: crane (bird, construction equipment)
- What is the difference between polysemy and homonymy?
 - Example: down
 - Homonyms: don't look **down** vs. a **down** coat
 - Polyseme: Bob went to the **bank** to deposit a check vs. the **bank** fired half of its staff

Synonymy and Antonymy

- Synonyms: words that have the same meaning in some or all contexts
 - couch/sofa, automobile/car
- Antonyms: senses that are opposite with respect to one feature of meaning, and otherwise very similar
 - long/short, fast/slow, hot/cold

Hyponymy and Hypernymy

- One sense is a hyponym of another if the first sense is more specific, denoting a subclass of the other
 - Y is a hyponym of X if every Y is a (kind of) X (*dog* is a hyponym of *canine*)
 - *car* is a hyponym of *vehicle*
 - *mango* is a hyponym of *fruit*
- Conversely hypernym
 - Y is a hypernym of X if every X is a (kind of) Y (*canine* is a hypernym of *dog*)
 - *vehicle* is a hypernym of *car*
 - *fruit* is a hypernym of *mango*
- *Coordinate terms*: Y is a coordinate term of X if X and Y share a hypernym (*wolf* is a coordinate term of *dog*, and *dog* is a coordinate term of *wolf*)

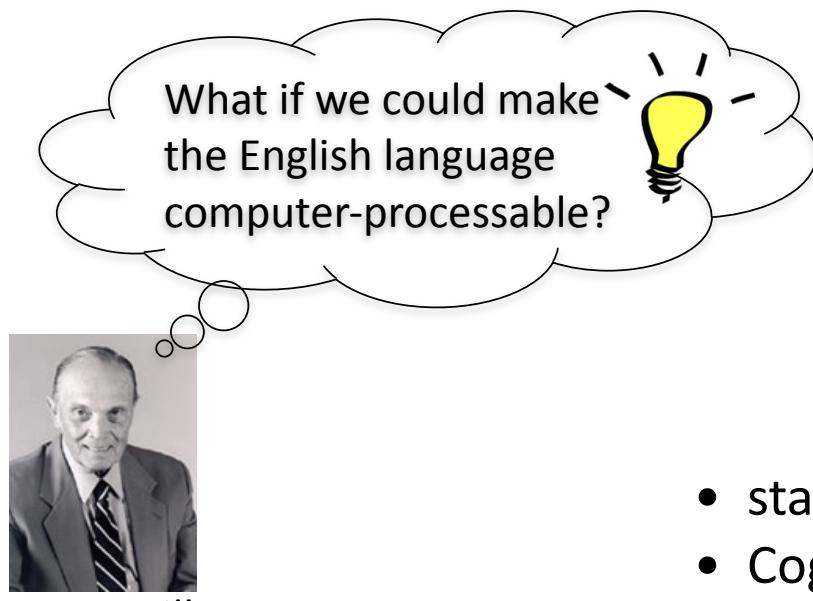
Meronymy and Holonymy

- Meronym: Y is a meronym of X if Y is a part of X
 - *window* is a meronym of *building*
- Holonym: Y is a holonym of X if X is a part of Y
 - *building* is a holonym of *window*

Verb Relations

- hypernym: the verb Y is a hypernym of the verb X if the activity X is a (kind of) Y
 - to perceive is an hypernym of to listen
- troponym: the verb Y is a troponym of the verb X if the activity Y is doing X in some manner
 - to lisp is a troponym of to talk
- entailment: the verb Y is entailed by X if by doing X you must be doing Y
 - to sleep is entailed by to snore
- coordinate terms: those verbs sharing a common hypernym
 - to lisp and to yell

WordNet



George Miller

- started in 1985
- Cognitive Science Laboratory, Princeton University
- written by lexicographers
- goal: support automatic text analysis and AI applications

[Miller, CACM 1995]

Semantic Relations in WordNet

- Nouns
 - Hypernyms, hyponyms, meronyms, holonyms, synonyms, antonyms
- Verbs
 - Hypernyms, troponyms, entailment
- Adjectives
 - Related noun, similar to, participle of verb
- Adverbs
 - Root adjectives

WordNet Semantic Relations

Relation	Meaning	Examples
Synonymy (N, V, Adj, Adv)	Same sense	(camera, photographic camera) (mountain climbing, mountaineering) (fast, speedy)
Antonymy (Adj, Adv)	Opposite	(fast, slow) (buy, sell)
Hypernymy (N)	Is-A	(camera, photographic equipment) (mountain climbing, climb)
Meronymy (N)	Part	(camera, optical lens) (camera, view finder)
Troponymy (V)	Manner	(buy, subscribe) (sell, retail)
Entailment (V)	X must mean doing Y	(buy, pay) (sell, give)

Hypernym Example

dog, domestic dog, *Canis familiaris*

=> canine, canid

=> carnivore

=> placental, placental mammal, eutherian

=> mammal, mammalian

=> vertebrate, craniate

=> chordate

=> animal, animate being, creature

=> organism, being

=> living thing

Hyponym Example

dog, domestic dog, Canis familiaris

=> puppy

=> pooch, doggie, doggy, barker, bow-wow

=> cur, mongrel, mutt

=> lapdog

=> toy dog, toy

=> hunting dog

=> working dog

WordNet size

Type	Number
#words	155k
#senses	117k
#word-sense pairs	207k
%words that are polysemous	17%
License	Proprietary, Free for research

<http://wordnet.princeton.edu/wordnet/man2.1/wnstats.7WN.html>

WordNet Limitations

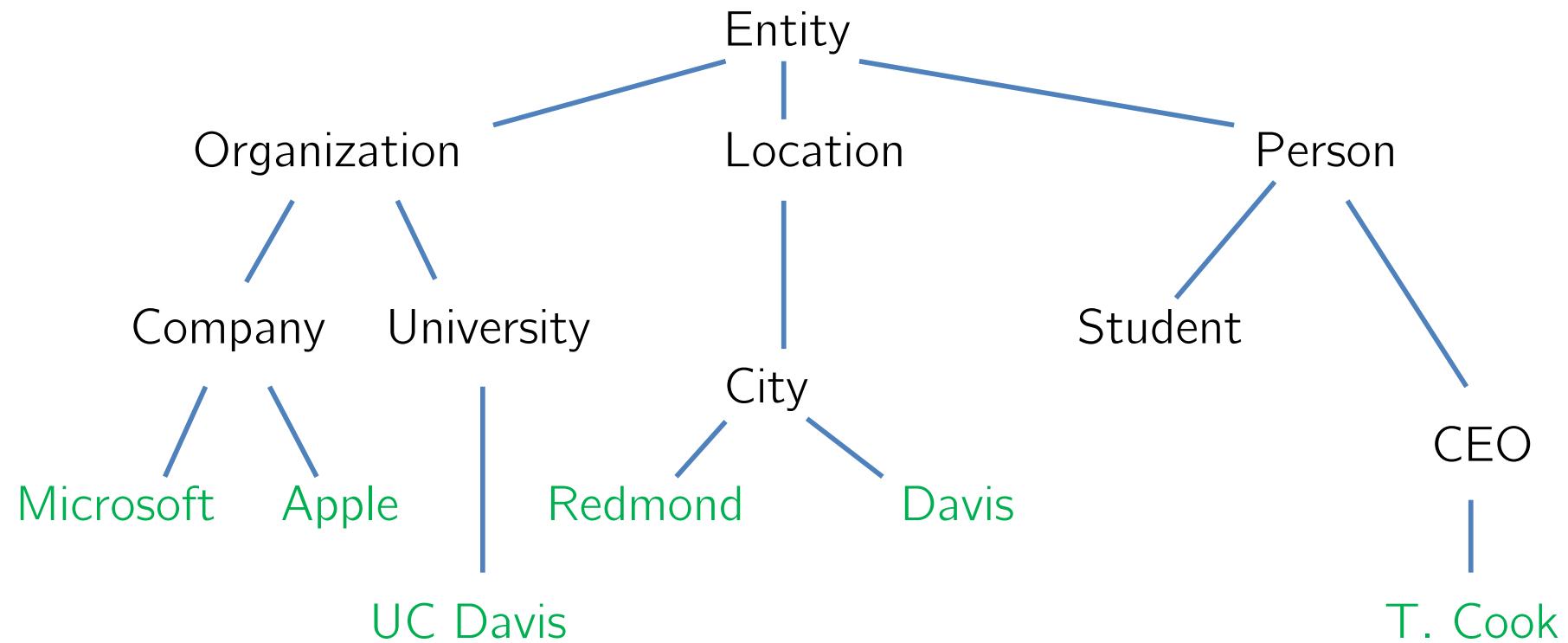
- Coverage
 - Many instances and classes missing
 - Example: The only computer scientist is Bill Gates!
 - Not all relations are listed
 - Does not cover all domains
 - Uneven coverage (more information on animals than people)

Information Extraction

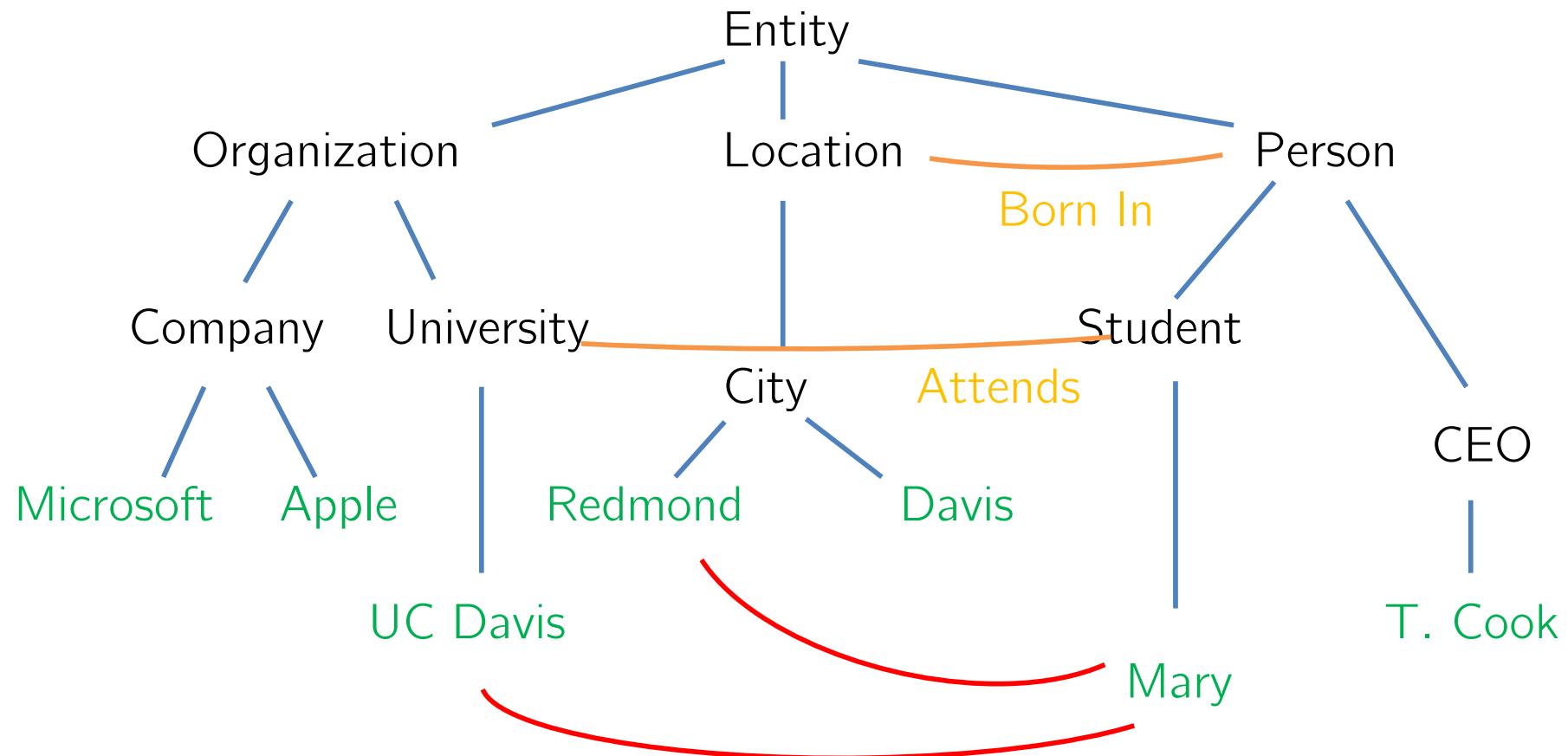
UC Davis LIN 127
Spring 2019

Kenji Sagae

Relations



Relations



Extracting knowledge from text

- Structured and unstructured data on the web
- Disorganized, inconsistent and constantly changing
- Automated Semantic Class Learning

Lexico-Syntactic Patterns

- “Even in the pre-inauguration months, the then President elect visited countries such as Venezuela, Brazil, Cuba, United States, France, Italy, and the United Kingdom.”
- <class> such as <instance₁>, <instance₂>, ..., and <instance_j>
- NP₀ such as NP₁{, NP₂, ... (and | or) NP_j} j ≥ 1

for all NP_i:

hyponym(NP_i, NP₀)

- hyponym(Venezuela, country)
- hyponym(Brazil, country)
- ...

Lexico-Syntactic Patterns

- such <class> as <instance₁>, <instance₂>, ... and <instance_j>
- such NP as {NP ,}* {(and | or)} NP
 - ... works by such authors as Herrick, Goldsmith, and Shakespeare
 - hyponym(author, Herrick)
 - hyponym(author, Goldsmith)
 - hyponym(author, Shakespeare)
- NP {, NP}* {,} or other NP
 - bruises, broken bones or other injuries

Sometimes it works

- Cities such as Los Angeles, Boston and Seattle
- He visited many countries such as France, Italy and Spain...
- Detailed information for several countries such as maps, currency, language...
- How can the reliability of a pattern be estimated?
- Good patterns should
 - Occur frequently in text
 - (Nearly) always suggest the relation of interest

Singly-anchored patterns

- A singly-anchored patterns has the name of the relation to be learned and one open slot (*) for the term to be harvested
 - relation *
 - * relation
- Submit to search engines, check returned snippets

Example: Learning City Names

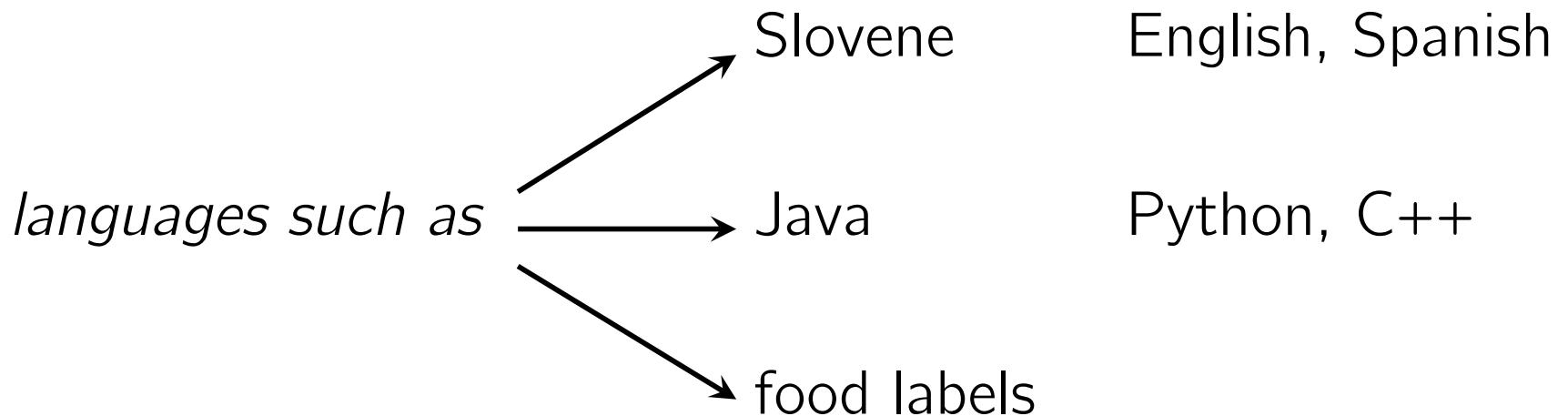
- Input: semantic class name
 - *City, town, cities, towns*
- We extract rules from Hearst (1992):
 - <class2> such as <NP_list>
 - <NP> is a <class1>
 - <class2> including <NP_list>
- Submit queries to search engine:
 - “cities such as”
 - “is a town”
 - “towns including”

Doubly-Anchored Patterns (DAP)

- **Doubly-anchored** pattern has anchoring either through conjunctions or terms
 - “*relation <seed> and **”
 - “*<seed> and * relation **”
 - “** relation * and <seed>*”
- *relation* is the name of the relation (i.e. “such as”, “fly to”, “work for”, “cause”)
- *<seed>* example term for the relation
- (*) indicates location of extracted terms

Advantages of DAP

- DAP drastically reduces ambiguity
 - Class names and class instances disambiguate each other



Limitations of DAP

- Sparsity when applied to small corpus
 - Typically used to extract knowledge from the web
- A single category member is insufficient for recall
 - Bootstrapping

Reckless* Bootstrapping

- Instantiate DAP with *ClassName* and one *<seed>* instance
- Feed the newly learned terms on *<seed>* position
- Conduct a breadth-first search

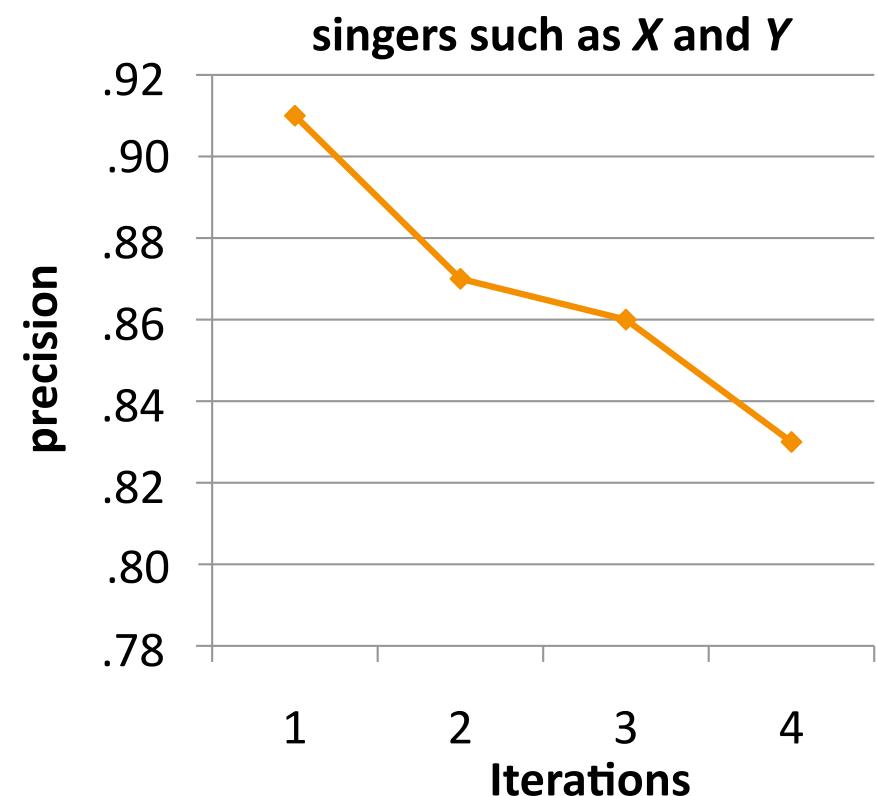
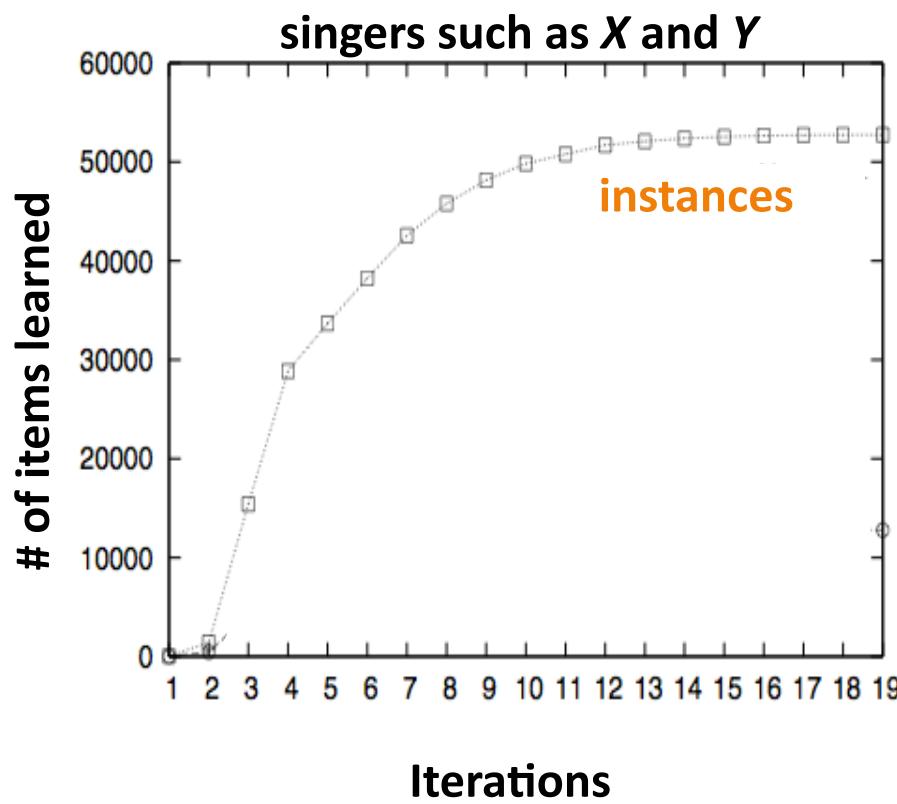
*states such as Alabama and **



Texas
Mississippi
Arkansas

*there is no check at any point

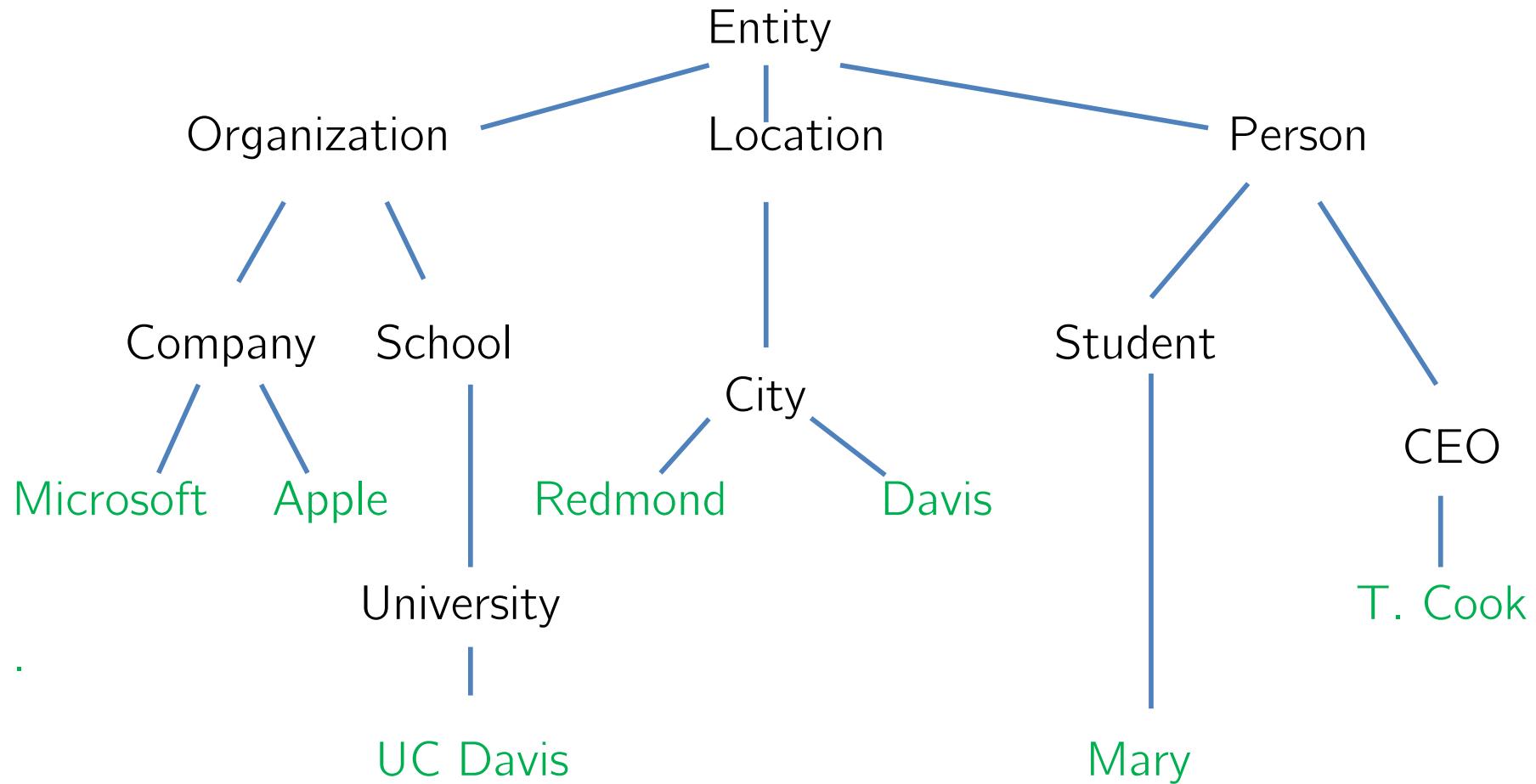
Learning Curves

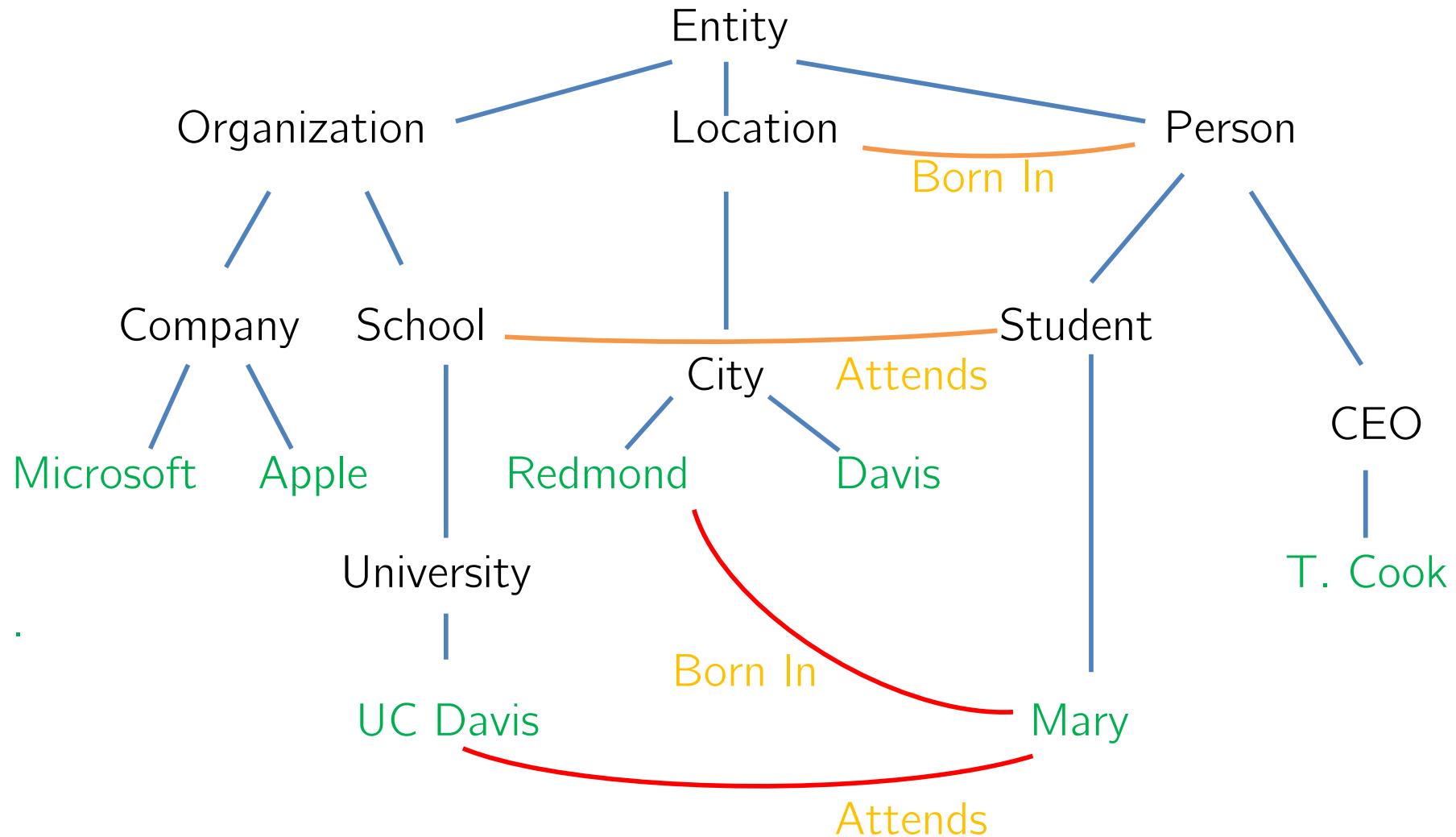


Pointwise Mutual Information, Word Vectors

UC Davis LIN 127
Spring 2019

Kenji Sagae





Recap:

You shall know a word by the company it keeps (J. R. Firth, 1957)

Quote now overused in NLP

Co-occurrence pattern of lexical items
words occurring together more frequently than by chance
(hostile takeover, associate with, Los Angeles)

Calculating co-occurrence

- Words occur together more often than by chance
 - Observed frequency of the two words together
 - Expected frequency of the two words together
- What types appear together most often?
 - of the
 - in the
 - ...

Pointwise Mutual Information

- Pointwise mutual information (PMI) compares:
 - Observed: the actual probability of the two words appearing together, $p(w_1, w_2)$
 - Expected: the probability of the two words appearing together if they are independent, $p(w_1)p(w_2)$
 - If we have two words, a and b:

$$pmi(a; b) \equiv \log \frac{p(a, b)}{p(a)p(b)} = \log \frac{p(a|b)}{p(a)} = \log \frac{p(b|a)}{p(b)}$$

- The higher the value, the more *surprising* it is

- (S1) mary drinks juice
- (S2) she likes apple juice
- (S3) mary eats apple
- (S4) mary drinks milk
- (S5) apple juice is sweet

P(juice) =

- (S1) mary drinks juice
- (S2) she likes apple juice
- (S3) mary eats apple
- (S4) mary drinks milk
- (S5) apple juice is sweet

$$P(\text{juice}) = 3/17$$

$$P(\text{juice} \mid \text{apple}) =$$

- (S1) mary drinks juice
- (S2) she likes apple juice
- (S3) mary eats apple
- (S4) mary drinks milk
- (S5) apple juice is sweet

$$P(\text{juice}) = 3/17$$

$$P(\text{juice} \mid \text{apple}) = 2/3$$

- (S1) mary drinks juice
- (S2) she likes apple juice
- (S3) mary eats apple
- (S4) mary drinks milk
- (S5) apple juice is sweet

$$P(\text{juice}) = 3/17$$

$$P(\text{juice} \mid \text{apple}) = 2/3$$

$$\text{PMI}(\text{apple}, \text{ juice})$$

$$= \log P(\text{juice} \mid \text{apple}) / P(\text{juice})$$

- (S1) mary drinks juice
- (S2) she likes apple juice
- (S3) mary eats apple
- (S4) mary drinks milk
- (S5) apple juice is sweet

$$P(\text{juice}) = 3/17$$

$$P(\text{juice} \mid \text{apple}) = 2/3$$

$$\text{PMI}(\text{apple}, \text{juice})$$

$$= \log P(\text{juice} \mid \text{apple}) / P(\text{juice})$$

$$= \log (2/3) / (3/17)$$

$$= 1.33$$

- (S1) mary drinks juice
- (S2) she likes apple juice
- (S3) mary eats apple
- (S4) mary drinks milk
- (S5) apple juice is sweet

$$P(\text{juice}) = 3/17$$

$$P(\text{juice} \mid \text{apple}) = 2/3$$

$$\text{PMI}(\text{apple}, \text{ juice})$$

$$= \log P(\text{juice} \mid \text{apple}) / P(\text{juice})$$

$$= \log (2/3) / (3/17)$$

$$= 1.33$$

$$P(\text{juice} \mid \text{drinks}) = 1/2$$

$$\text{PMI}(\text{drinks}, \text{ juice})$$

$$= \log (1/2) / (3/17)$$

$$= 1.04$$

Word Meaning

- What is the meaning of a word?
- What does it mean for words to be similar?
- What does it mean for words to be related?
 - Syntactically
 - Semantically
 - ...

- desk / desk
- desk / desks
- desk / table
- desk / chair
- desk / tables
- desk / furniture
- desk / office
- desk / road
- desk / fish
- desk / looked
- desk / between

Vectors

- What are vectors?
- Dimensions?
 - [3]
 - [3, 2]
 - [3, 2, 7]
 - ...

- Student vector representation
 - Dimensions?
 - Major
 - Year
 - Courses taken
 - ...
- Movie vector representation
 - Dimensions?
 - Genre
 - Director
 - Year
 - ...

Which students are similar?

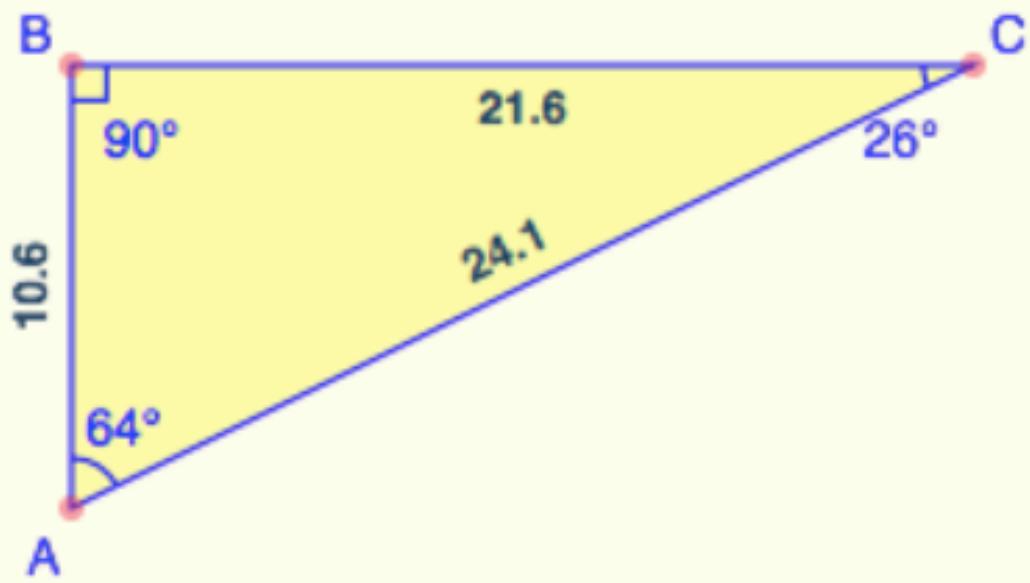
- Intuitively, students with similar attributes are similar
- How do we quantify similarity in a vector space?
- Cosine similarity (one of many ways)

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$



$$\cos A = \frac{17.4}{18.3} = 0.951$$

$$\cos C = \frac{5.8}{18.3} = 0.317$$

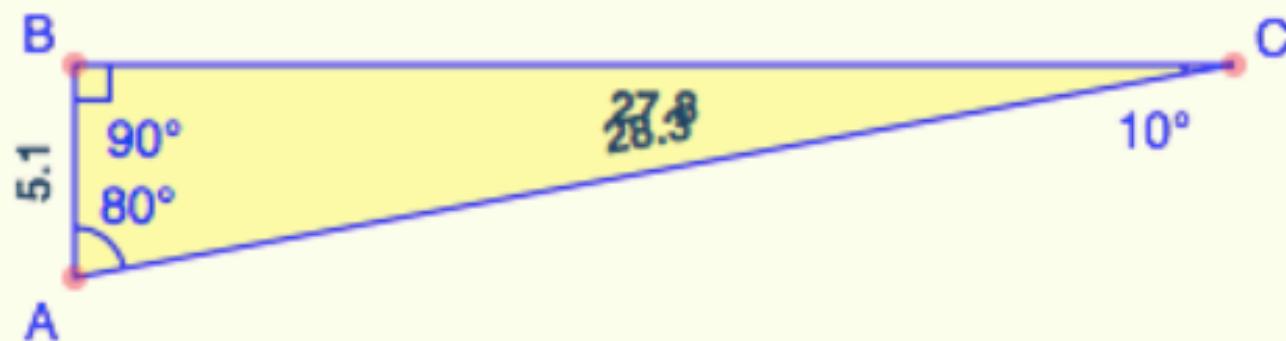


$$\cos A = \frac{10.6}{24.1} = 0.440$$

$$\cos C = \frac{21.6}{24.1} = 0.896$$

$$\cos A = \frac{5.1}{28.3} = 0.180$$

$$\cos C = \frac{27.8}{28.3} = 0.982$$



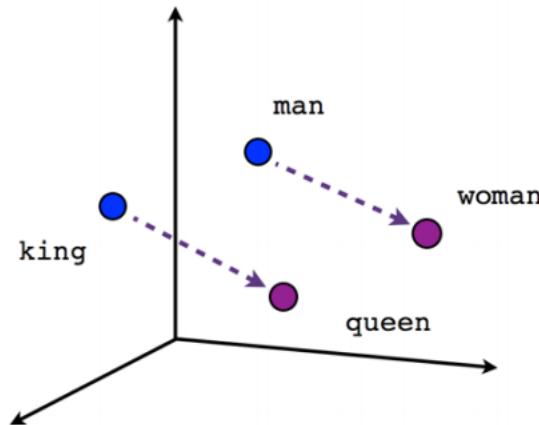
Vector Representations

- From discrete to distributed and continuous
- Each dimension contributes to “meaning” in a different way
- Mature mathematical framework
- Motivated theoretically and practically

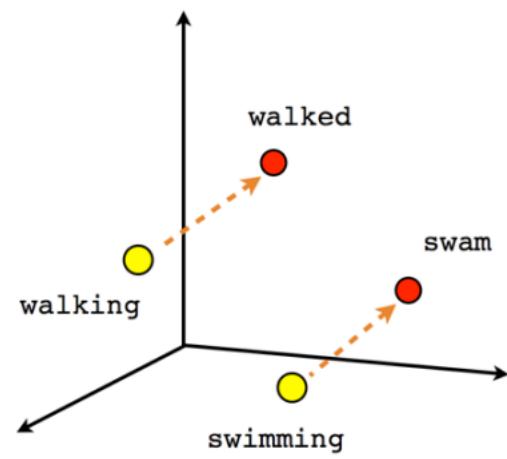
Word Vector Representations

- Distributional semantics
 - Items with similar distributions have similar meanings
 - Distributional hypothesis
- Each word is represented as a vector
 - Possibly with many dimensions
 - 30, 100, 300, 1M, more
 - desk = [0.234, 0.657, -1.345, 0.986, 1.245]
 - table = [0.256, 0.896, -0.345, 1.024, 0.234]
 - fish = [1.753, -2.451, 1.234, -1.234, 1.934]

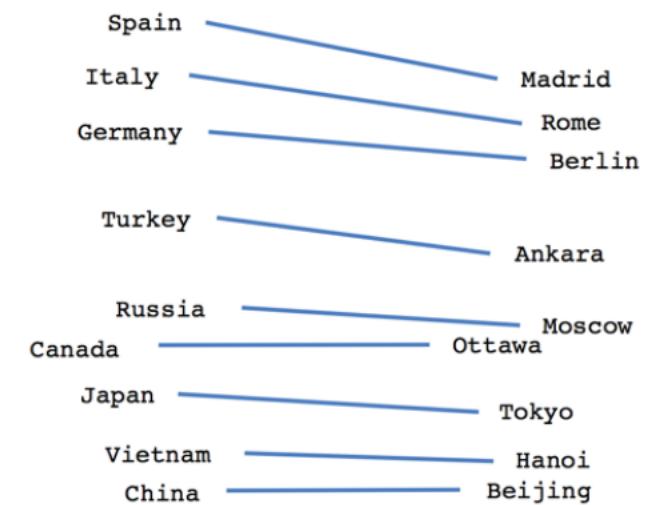




Male-Female



Verb tense



Country-Capital

What are the dimensions in word vectors?

- Co-occurrence counts
 - What does it mean for words to co-occur
 - How do we count co-occurrence
- Similarity measures
- (Also, dimensionality reduction)

(Using previous word as context)

(S1) mary drinks juice

(S2) apple juice

(S3) mary likes apple juice

	mary	apple	drinks	likes	juice	< s >
mary	0	0	0	0	0	2
apple	0	0	0	1	0	1
drinks	1	0	0	0	0	0
likes	1	0	0	0	0	0
juice	0	1	1	0	0	0

(Using previous and next words as context)

(S1) mary drinks juice

(S2) apple juice

(S3) mary likes apple juice

mary apple drinks likes juice <s> mary2 apple2 drinks2 likes2 juice2 </s>

mary 0 0 0 0 0 2 0 0 1 1 0 0

apple 0 0 0 1 0 1 0 0 0 0 1 0

drinks 1 0 0 0 0 0 0 0 0 1 0

likes 1 0 0 0 0 0 1 0 0 0 0

juice 0 1 1 0 0 0 0 0 0 0 0 3

Term-document matrix

(D1) mary drinks juice

(D2) apple juice

(D3) mary likes apple juice

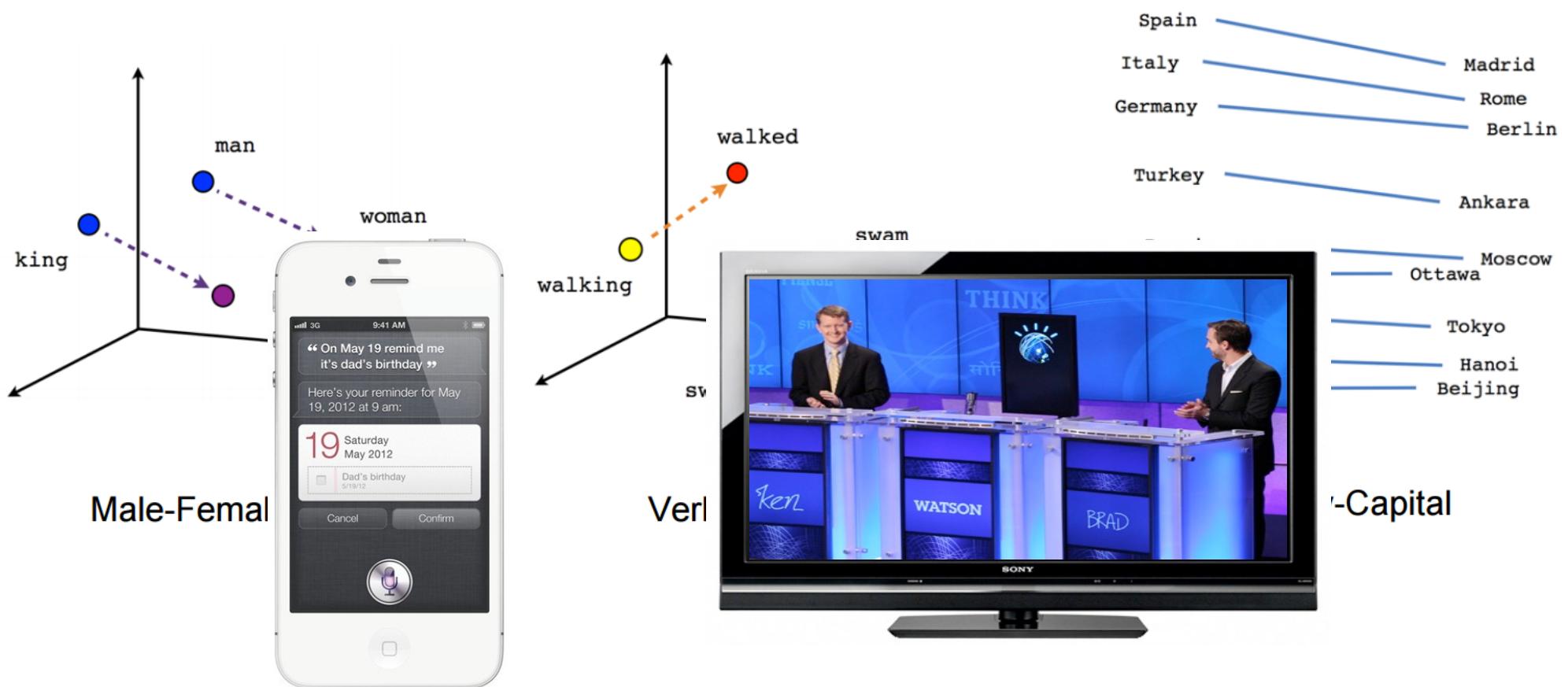
mary apple drinks likes juice

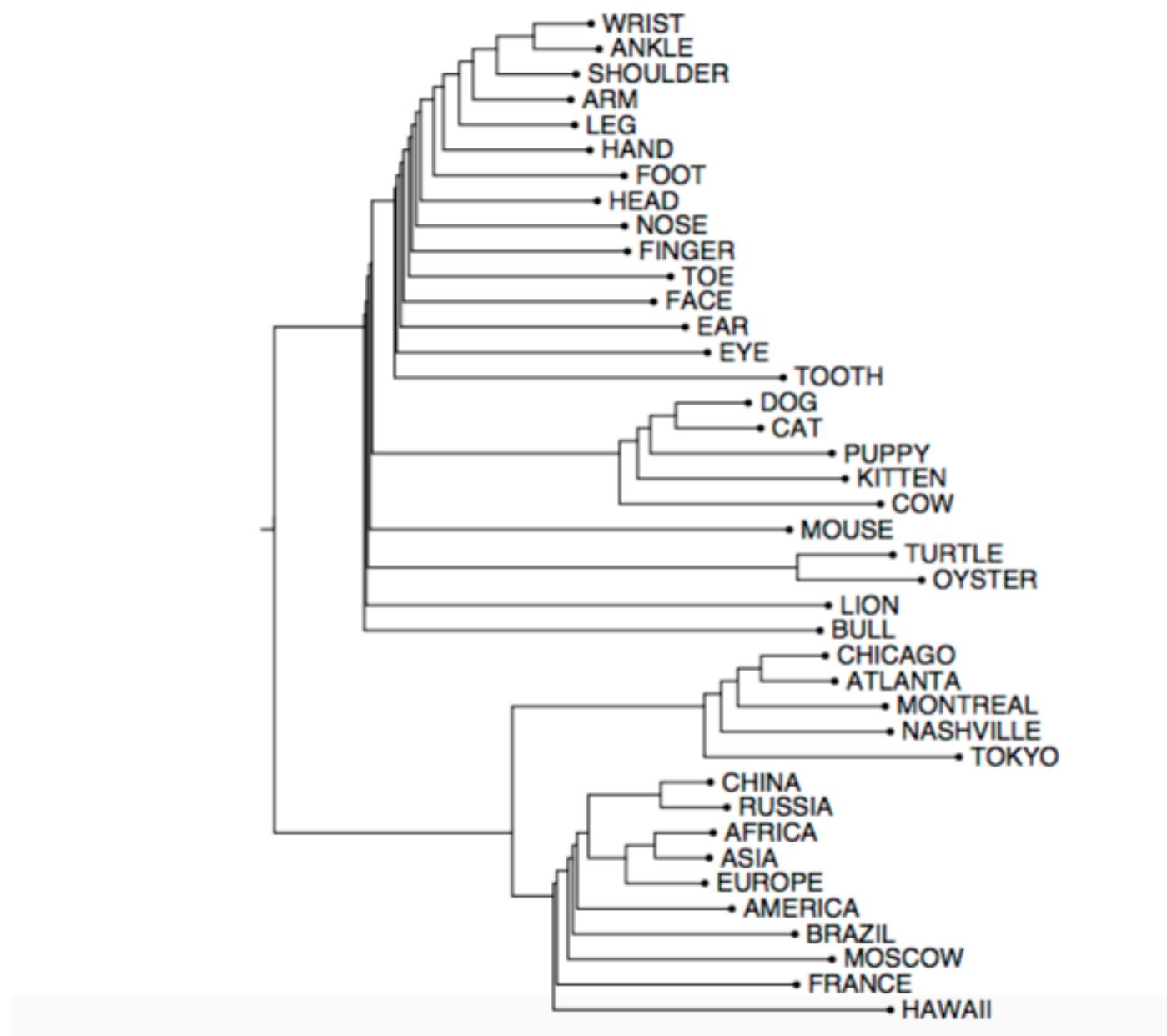
D1	1	0	1	0	1
D2	0	1	0	0	1
D3	1	1	0	1	1

Explicit Word Vectors

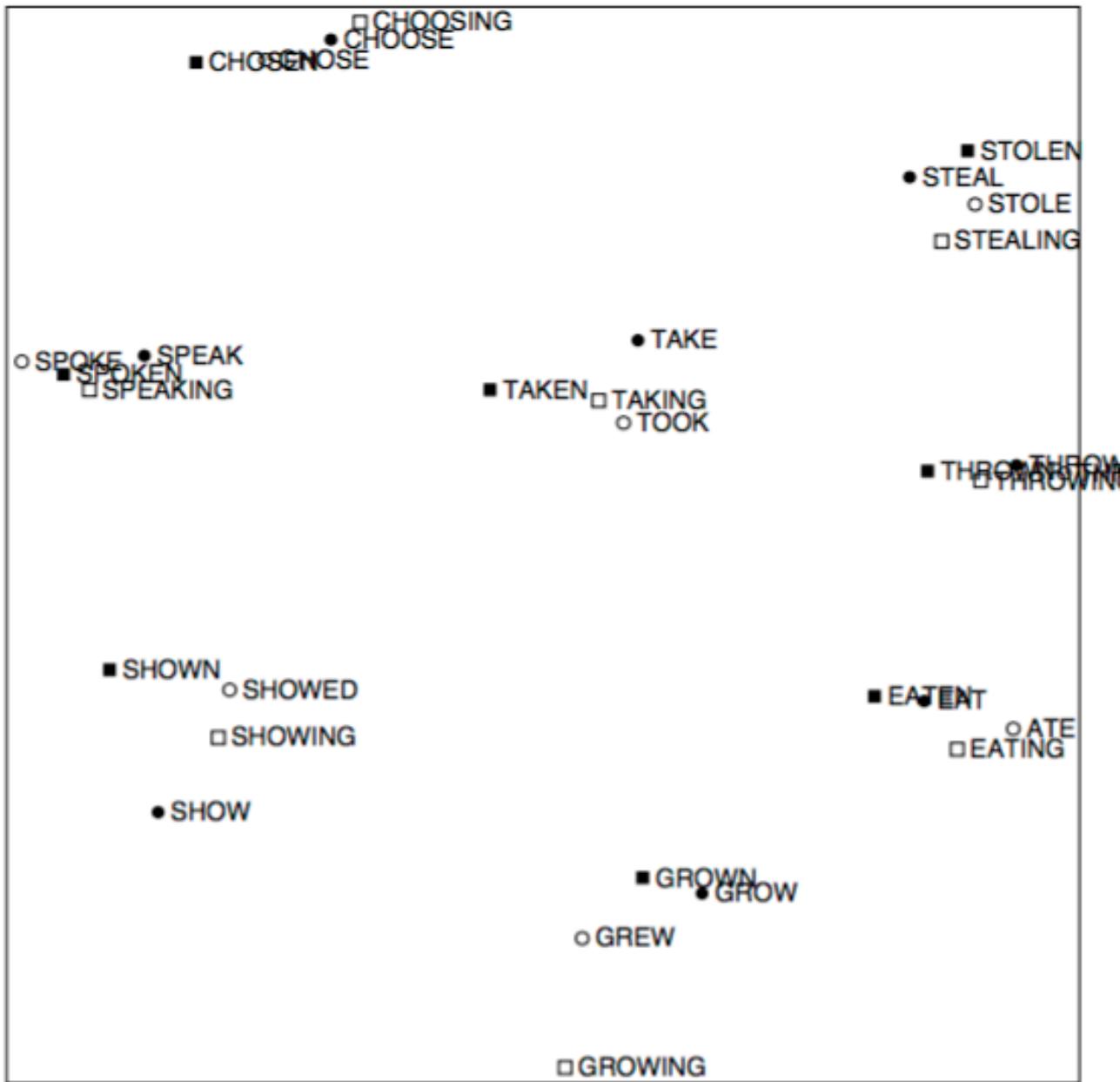
- Context window
 - E.g. Two words before, two words after
 - Position of word in context is marked explicitly
 - red_p1, red_p2, red_n1, red_n2
- Vector dimensionality is large
 - $4 |V|$
- Values: Pointwise mutual information

- Word embeddings, word vectors
 - <https://code.google.com/archive/p/word2vec/>
 - <http://nlp.stanford.edu/projects/glove/>
 - <https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/>
- Domain and genre matter

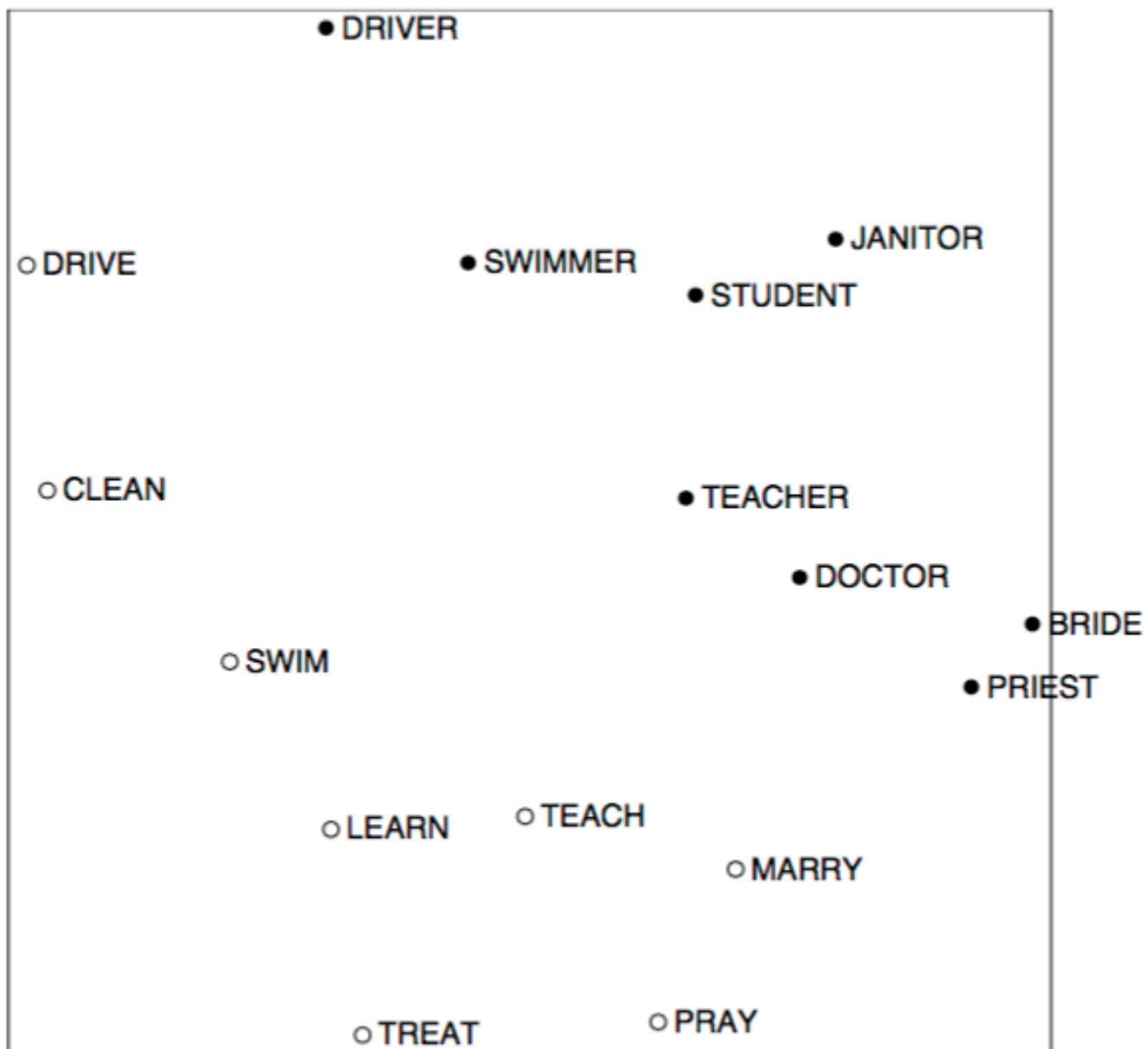


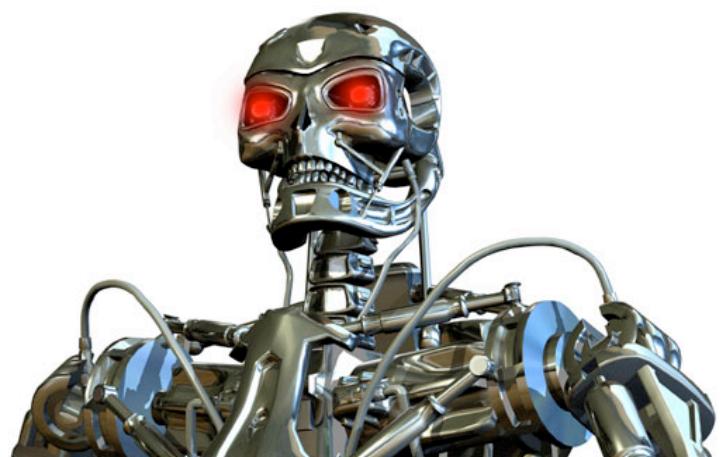
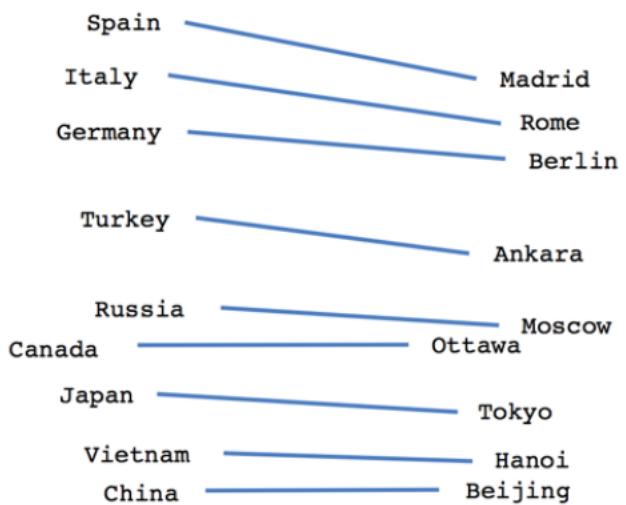
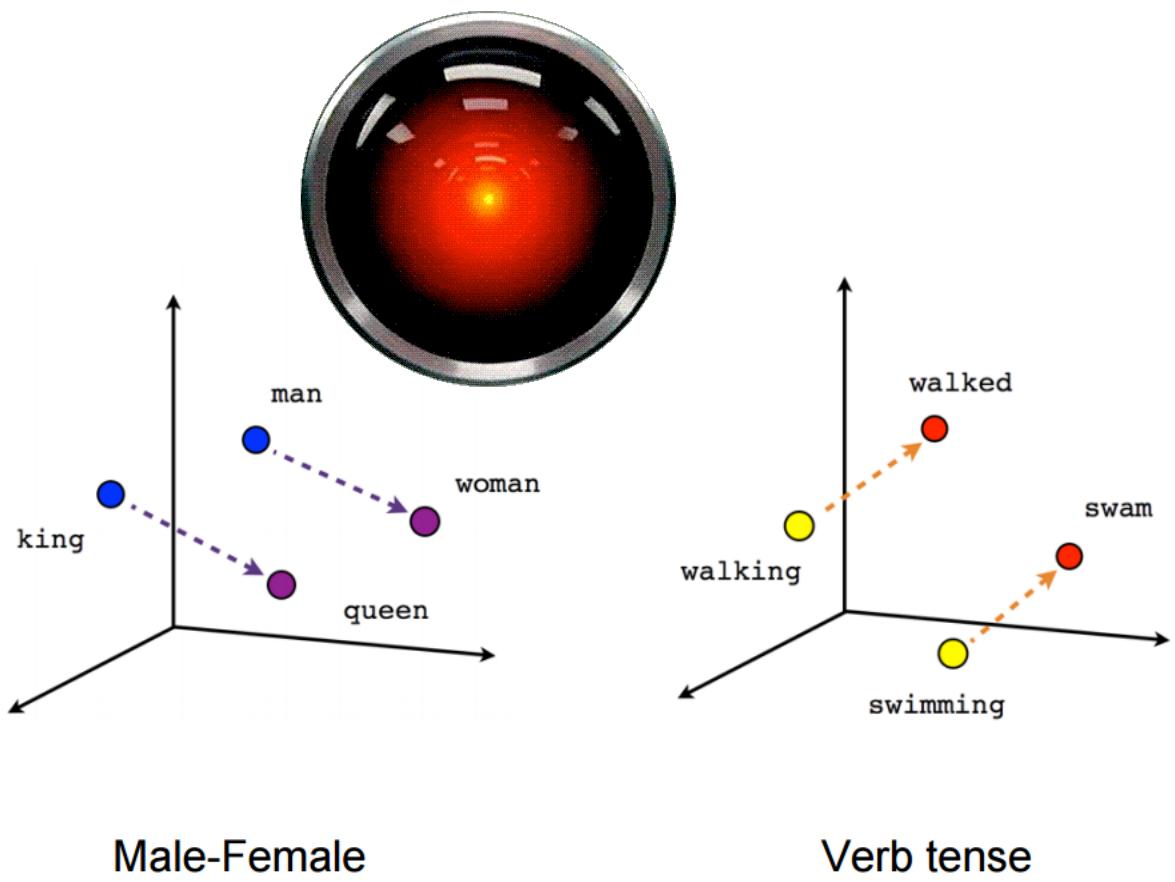


Rohde et al. 2005



Rohde et al. 2005





Language Modeling

UC Davis LIN 127
Spring 2019

Kenji Sagae

Bias and Ethical AI

- LinkedIn name suggestions
- Personalized search
- Microsoft Tay
- Voice-enabled car systems
- Predictive policing
- Discriminatory advertising

- Is there a problem?
 - Why, or why not?

Language Modeling

Speech Recognition

(oversimplified summary)

- Noisy channel model
- Predict a sentence (word sequence) given the acoustics

$$\arg \max_{\text{wordsequence}} P(\text{wordsequence} \mid \text{acoustics}) =$$

$$\arg \max_{\text{wordsequence}} \frac{P(\text{acoustics} \mid \text{wordsequence}) \times P(\text{wordsequence})}{P(\text{acoustics})}$$

$$\arg \max_{\text{wordsequence}} P(\text{acoustics} \mid \text{wordsequence}) \times P(\text{wordsequence})$$

Probabilistic Language Models

- Given a sentence (word sequence) x , find $P(x)$
 - $P(\text{I ate a sandwich}) \gg P(\text{a ate I sandwich})$
- Decompose x into small pieces
 - Uses the $n-1$ words to predict the next one
 - Unigram: look only at individual words
 - Bigram: pair of adjacent words
 - Trigram: contiguous sequence of three words
 - Why?