# Natural Language Processing

## Logistic Regression

Yulia Tsvetkov

yuliats@cs.washington.edu

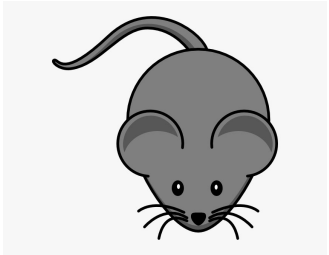
PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

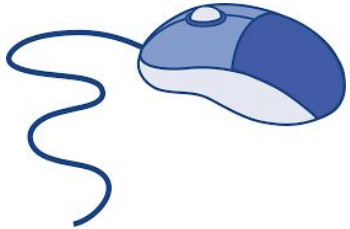
# Back to the introduction topics…

# Why is language interpretation hard?

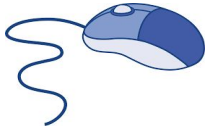
1. Ambiguity
2. Scale
3. Variation
4. Sparsity
5. Expressivity
6. Unmodeled variables
7. Unknown representation $\mathcal{R}$

# Ambiguity: word sense disambiguation





# Ambiguity



- Ambiguity at multiple levels:
  - Word senses: **bank** (finance or river?)
  - Part of speech: **chair** (noun or verb?)
  - Syntactic structure: **I can see a man with a telescope**
  - Multiple: **I saw her duck**

# Semantic analysis

- Every language sees the world in a different way
  - For example, it could depend on cultural or historical conditions


  - Russian has very few words for colors, Japanese has hundreds
  - Multiword expressions, e.g. happy as a clam, it's raining cats and dogs or wake up and metaphors, e.g. love is a journey are very different across languages

# Why is language interpretation hard?

1. Ambiguity
2. Scale
3. Variation
4. Sparsity
5. Expressivity
6. Unmodeled variables
7. Unknown representation $\mathcal{R}$

# Scale

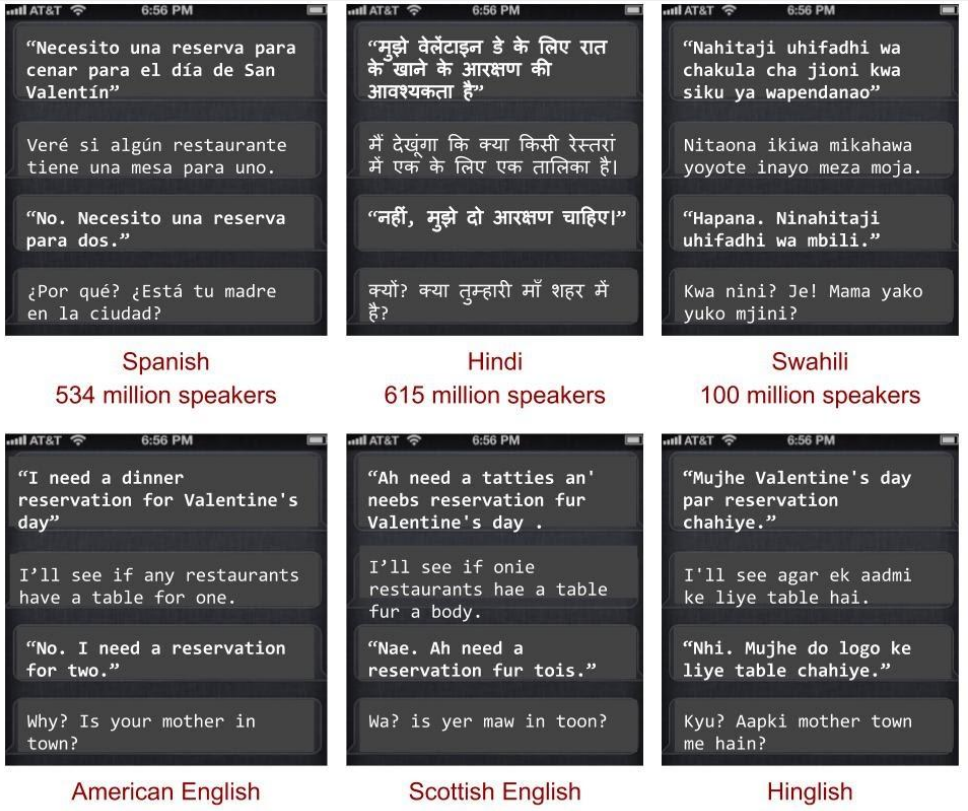
- ~7K languages
- Thousands of language varieties


Englishes

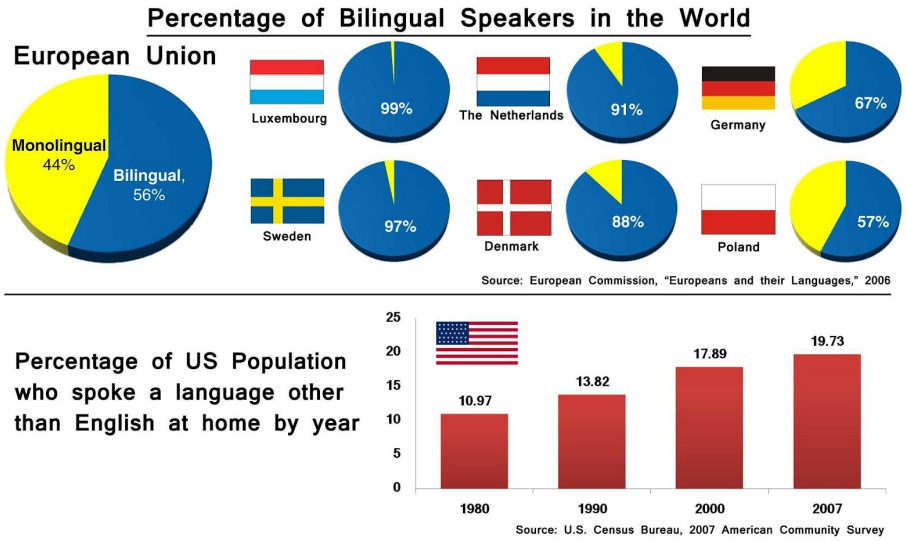

Africa is a continent with a very high linguistic diversity: there are an estimated 1.5-2K African languages from 6 language families. 1.33 billion people

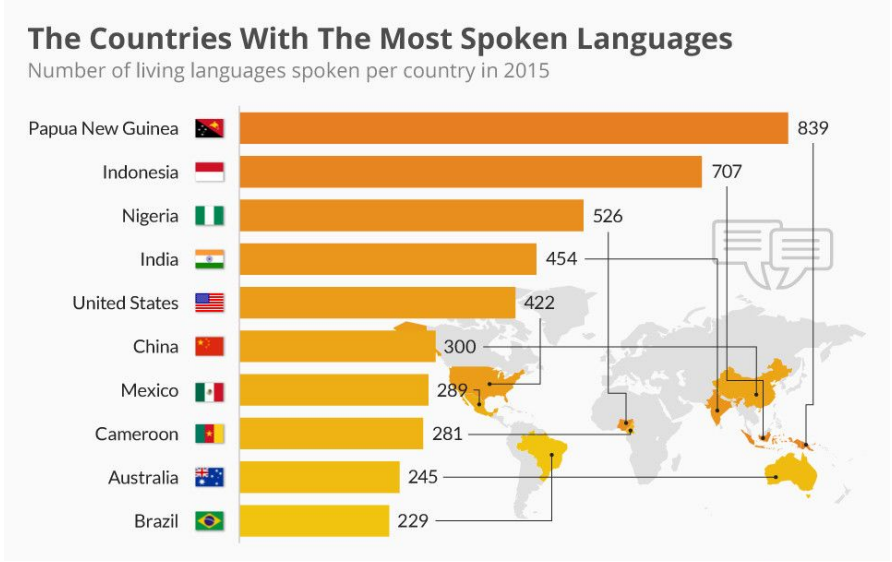
# NLP beyond English

- ~7,000 languages
- thousands of language varieties



# Most of the world today is multilingual




Source: US Census Bureau

Source: Ethnologue

# Tokenization

这是一个简单的句子

WORDS This is a simple sentence

זה משפט פשוט

# Tokenization + disambiguation

| | |
|---|---|
| in tea | בתה |
| her daughter | |

- most of the vowels unspecified

| | |
|---|---|
| in tea | בתה |
| in the tea | בהתה |
| that in tea | שבתה |
| that in the tea | שבהתה |
| and that in the tea | ושבהתה |
| | ושבתה |
| and her saturday | ו+שבת+ה |
| and that in tea | ו+ש+ב+תה |
| and that her daughter | ו+ש+בת+ה |

- most of the vowels unspecified
- particles, prepositions, the definite article, conjunctions attach to the words which follow them
- tokenization is highly ambiguous

# Tokenization + morphological analysis

- Quechua

Much'ananayakapushasqakupuniñataqsunamá

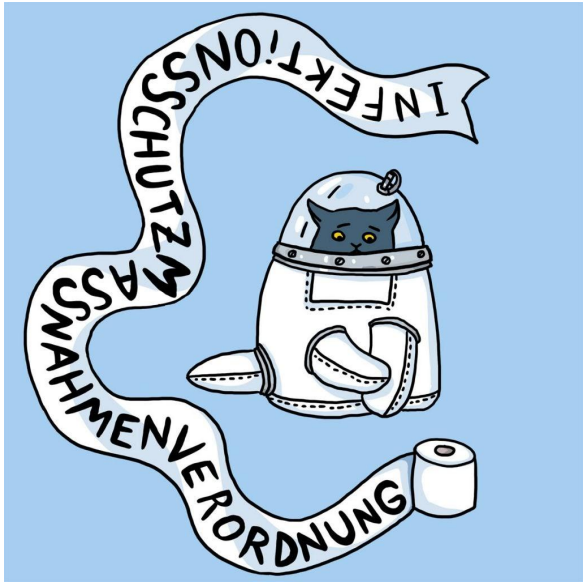
Much'a -na -naya -ka -pu -sha -sqa -ku -puni -ña -taq -suna -má

*"So they really always have been kissing each other then"*

| | |
|---|---|
| Much'a | to kiss |
| -na | expresses obligation, lost in translation |
| -naya | expresses desire |
| -ka | diminutive |
| -pu | reflexive (kiss *eachother*) |
| -sha | progressive (kiss*ing*) |
| -sqa | declaring something the speaker has not personally witnessed |
| -ku | 3rd person plural (they kiss) |
| -puni | definitive (really*) |
| -ña | always |
| -taq | statement of contrast (...then) |
| -suna | expressing uncertainty (So...) |
| -má | expressing that the speaker is surprised |

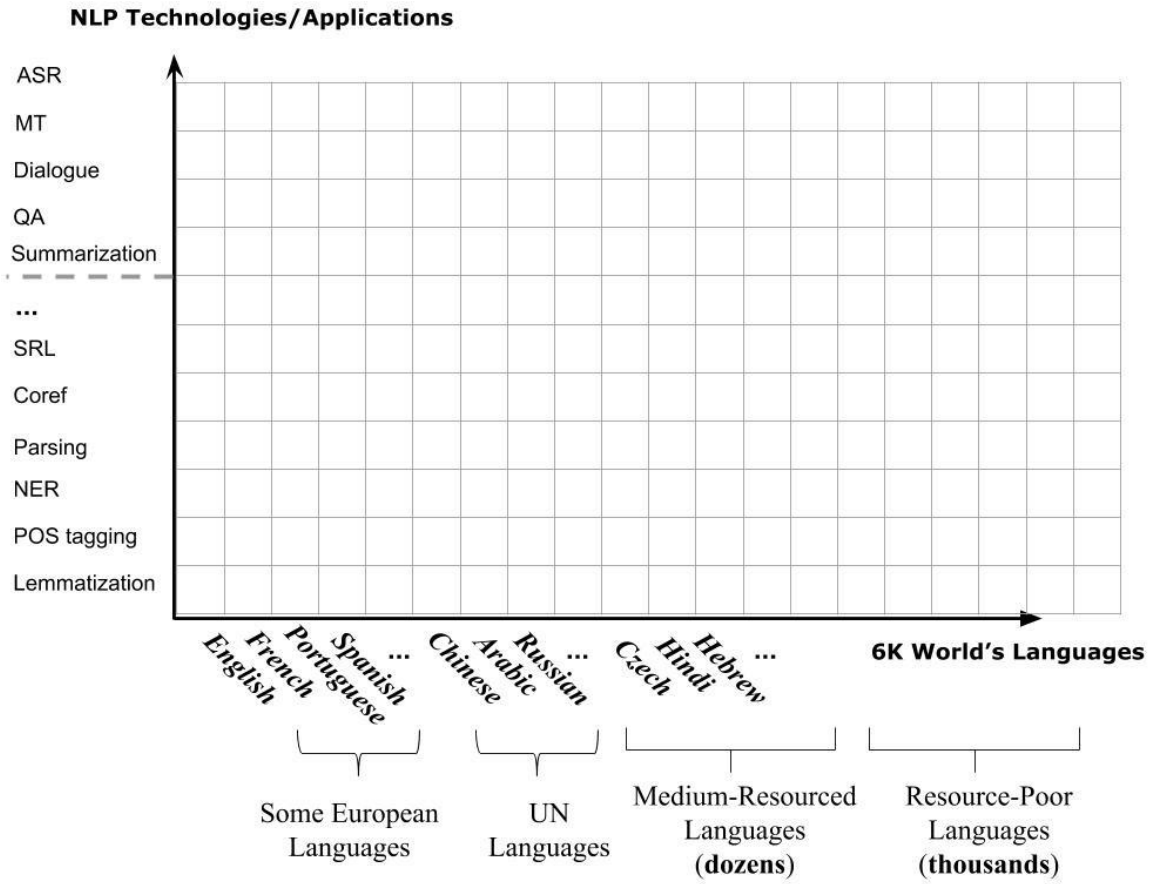
# Tokenization + morphological analysis

- German




Infektionsschutzmaßnahmenverordnung

“Infection Protection Measures Ordinance”


NLP Technologies/Applications
ASR
MT
Dialogue
QA
Summarization
...
SRL
Coref
Parsing
NER
POS tagging
Lemmatization
English
French
Portuguese
Spanish
...
Chinese
Arabic
Russian
...
Czech
Hindi
Hebrew
...
6K World's Languages
Some European Languages
UN Languages
Medium-Resourced Languages (dozens)
Resource-Poor Languages (thousands)

# Why is language interpretation hard?

1. Ambiguity
2. Scale
3. Variation
4. Sparsity
5. Expressivity
6. Unmodeled variables
7. Unknown representation $\mathcal{R}$

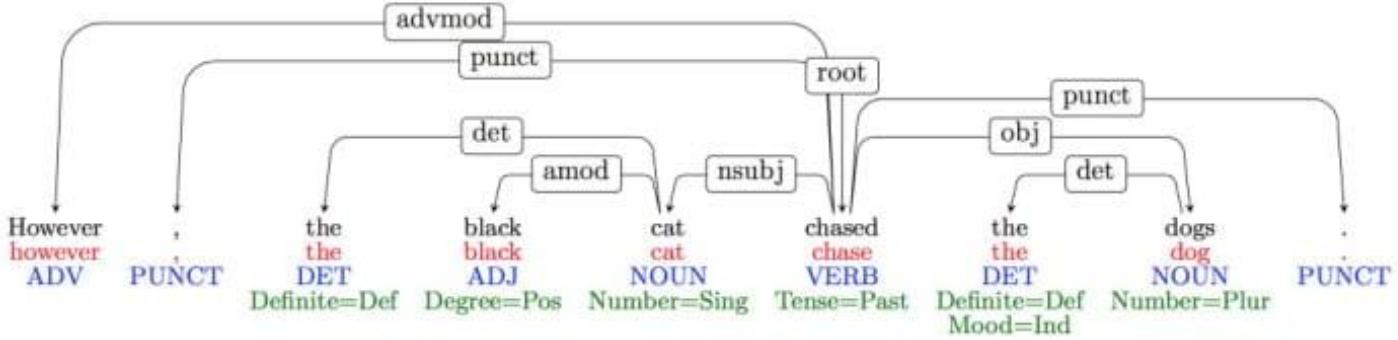
# Linguistic variation

- Non-standard language, emojis, hashtags, names


**chowdownwithchan** #crab and #pork #xiaolongbao at @dintaifungusa... where else? 😂 🤷‍♀️ Note the cute little crab indicator in the 2nd pic 🦀 💕

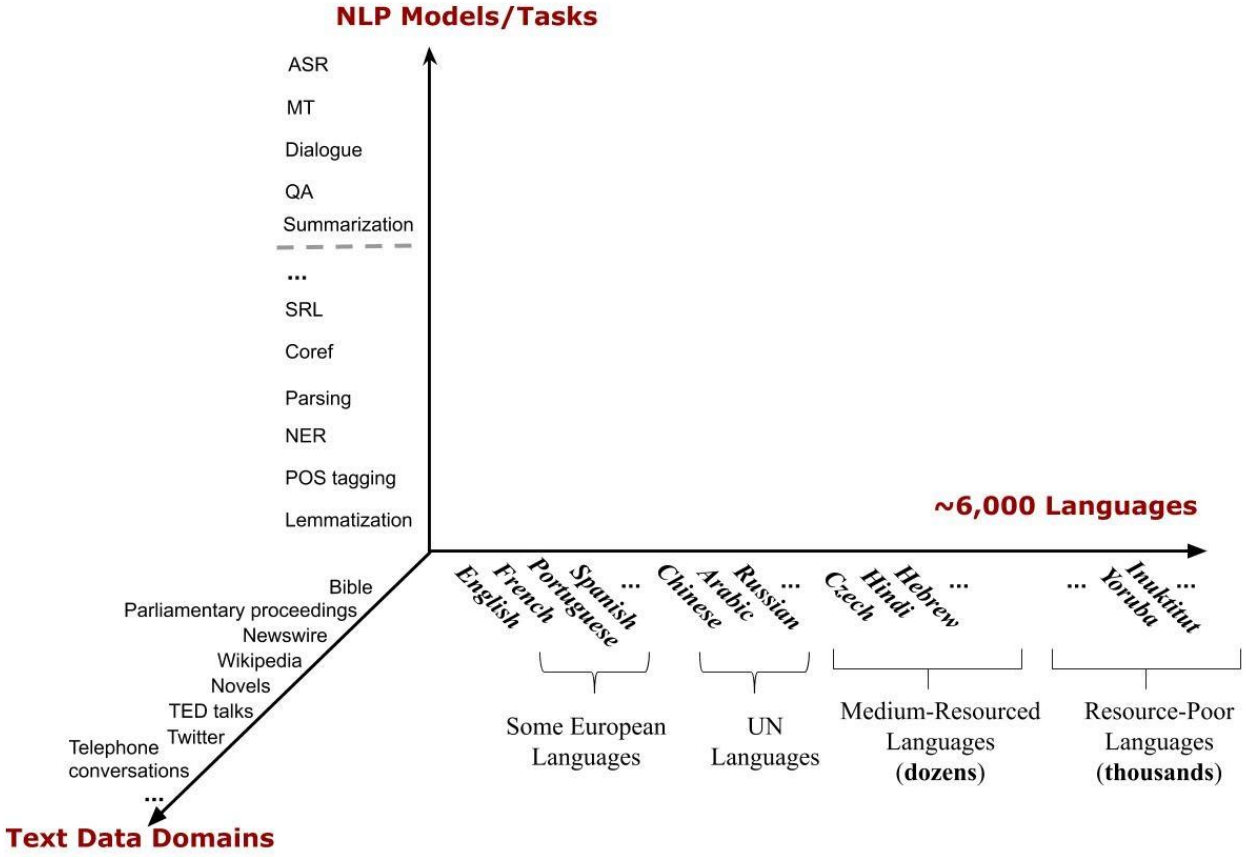
# Variation

- Suppose we train a part of speech tagger or a parser on the Wall Street Journal




- What will happen if we try to use this tagger/parser for social media??

@_rkpntrnte hindi ko alam babe eh, absent ako kanina I'm sick rn hahaha 😌🙌


NLP Models/Tasks
ASR
MT
Dialogue
QA
Summarization
...
SRL
Coref
Parsing
NER
POS tagging
Lemmatization
~6,000 Languages
English
French
Spanish
Portuguese
...
Chinese
Arabic
Russian
...
Czech
Hindi
Hebrew
...
...
Yoruba
Inuktitut
...
Some European Languages
UN Languages
Medium-Resourced Languages (dozens)
Resource-Poor Languages (thousands)
Bible
Parliamentary proceedings
Newswire
Wikipedia
Novels
TED talks
Twitter
Telephone conversations
...
Text Data Domains

# Why is language interpretation hard?

1. Ambiguity
2. Scale
3. Variation
4. Sparsity
5. Expressivity
6. Unmodeled variables
7. Unknown representation $\mathcal{R}$

# Sparsity

Sparse data due to Zipf's Law

- To illustrate, let's look at the frequencies of different words in a large text corpus
- Assume "word" is a string of letters separated by spaces

# Word Counts

Most frequent words in the English Europarl corpus (out of 24m word tokens)

| any word | | nouns | |
|---|---|---|---|
| Frequency | Token | Frequency | Token |
| 1,698,599 | the | 124,598 | European |
| 849,256 | of | 104,325 | Mr |
| 793,731 | to | 92,195 | Commission |
| 640,257 | and | 66,781 | President |
| 508,560 | in | 62,867 | Parliament |
| 407,638 | that | 57,804 | Union |
| 400,467 | is | 53,683 | report |
| 394,778 | a | 53,547 | Council |
| 263,040 | I | 45,842 | States |

# Word Counts

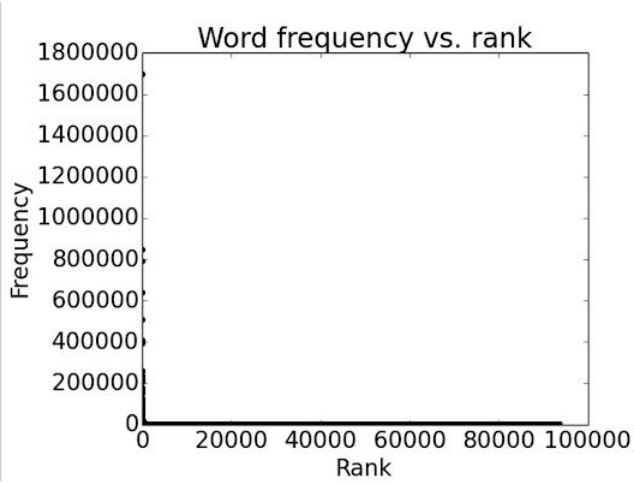
But also, out of 93,638 distinct words (word types), 36,231 occur only once.

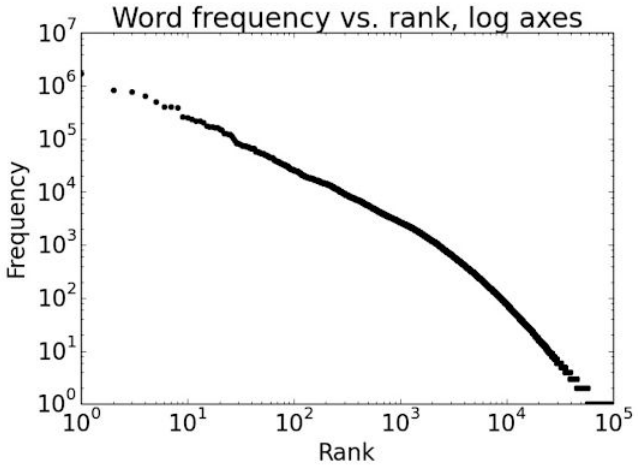
Examples:

- cornflakes, mathematicians, fuzziness, jumbling
- pseudo-rapporteur, lobby-ridden, perfunctorily,
- Lycketoft, UNCITRAL, H-0695
- policyfor, Commissioneris, 145.95, 27a

# Plotting word frequencies

Order words by frequency. What is the frequency of nth ranked word?

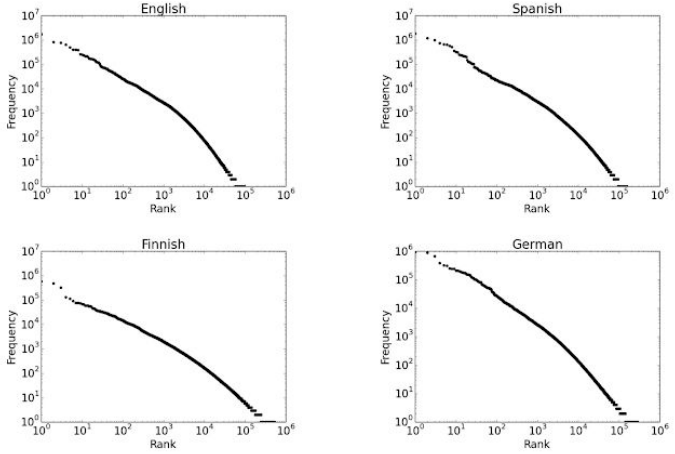


# Zipf's Law

## Implications

- Regardless of how large our corpus is, there will be a lot of infrequent (and zero-frequency!) words
- This means we need to find clever ways to estimate probabilities for things we have rarely or never seen



# Why is language interpretation hard?

1. Ambiguity
2. Scale
3. Variation
4. Sparsity
5. Expressivity
6. Unmodeled variables
7. Unknown representation $\mathcal{R}$

# Expressivity

Not only can one form have different meanings (ambiguity) but the same meaning can be expressed with different forms:

She gave the book to Tom vs. She gave Tom the book

Some kids popped by vs. A few children visited

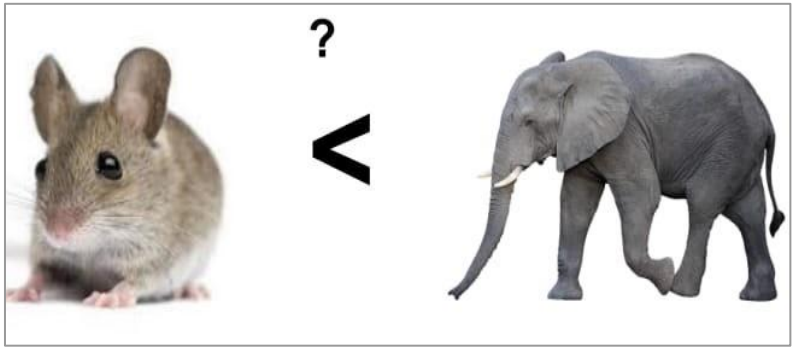
Is that window still open? vs. Please close the window

# Why is language interpretation hard?

1. Ambiguity
2. Scale
3. Variation
4. Sparsity
5. Expressivity
6. Unmodeled variables
7. Unknown representation $\mathcal{R}$

# Unmodeled variables

"Drink this milk"




## World knowledge

- I dropped the glass on the floor and it broke
- I dropped the hammer on the glass and it broke

# Why is language interpretation hard?

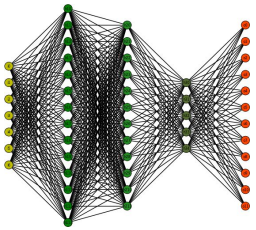
1. Ambiguity
2. Scale
3. Variation
4. Sparsity
5. Expressivity
6. Unmodeled variables
7. Unknown representation $\mathcal{R}$

# Unknown representation

- Very difficult to capture what is $\mathcal{R}$, since we don't even know how to represent the knowledge a human has/needs:
  - What is the "meaning" of a word or sentence?
  - How to model context?
  - Other general knowledge?

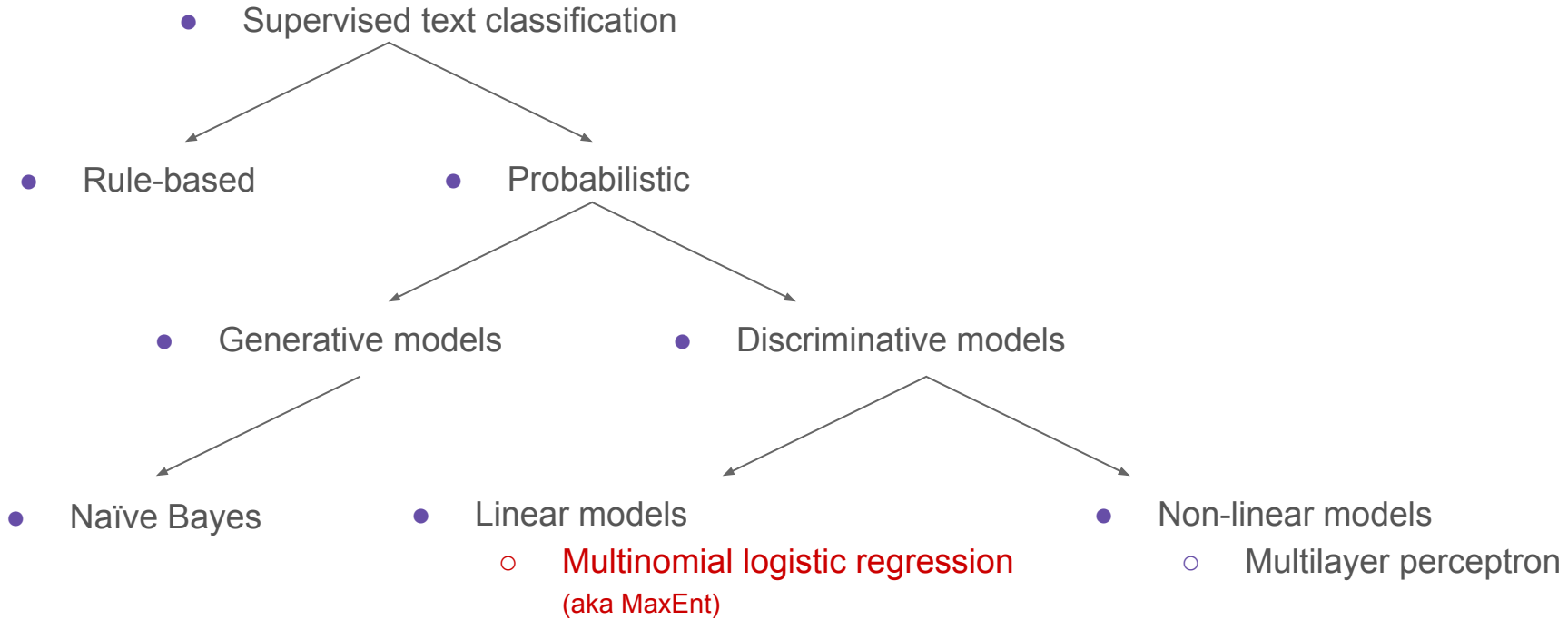
# Dealing with ambiguity

- How can we model ambiguity and choose the correct analysis in context?
  - non-probabilistic methods (FSMs for morphology, CKY parsers for syntax) return *all possible analyses*.
  - probabilistic models (HMMs for part-of-speech tagging, PCFGs for syntax) and algorithms (Viterbi, probabilistic CKY) return *the best possible analysis,* i.e., the most probable one according to the model
  - Neural networks, pretrained language models now provide end-to-end solutions



- But the "best" analysis is only good if our probabilities are accurate. Where do they come from?

# Next class: Logistic regression



# Readings

- J&M Chapter 5 https://web.stanford.edu/~jurafsky/slp3/5.pdf

# Logistic regression classifier

- Important analytic tool in natural and social sciences
- Baseline supervised machine learning tool for classification
- Is also the foundation of neural networks

# Text classification

Input:

- a document $d$ (e.g., a movie review)
- a fixed set of classes $C = \{c_1, c_2, \ldots c_j\}$ (e.g., positive, negative, neutral)

Output

- a predicted class $\hat{y} \in C$

# Binary classification in logistic regression

- Given a series of input/output pairs:
  - $(x^{(i)}, y^{(i)})$

- For each observation $x^{(i)}$
  - We represent $x^{(i)}$ by a feature vector $\{x_1, x_2, \ldots, x_n\}$
  - We compute an output: a predicted class $\hat{y}^{(i)} \in \{0,1\}$

# Features in logistic regression

- For feature $x_i \in \{x_1, x_2, \ldots, x_n\}$, weight $w_i \in \{w_1, w_2, \ldots, w_n\}$ tells us how important is $x_i$
  - $x_i$ = "review contains 'awesome'": $w_i$ = +10
  - $x_j$ = "review contains horrible": $w_j$ = -10
  - $x_k$ = "review contains 'mediocre'": $w_k$ = -2

# Logistic Regression for one observation x

- Input observation: vector $x^{(i)} = \{x_1, x_2, \ldots, x_n\}$
- Weights: one per feature: $W = [w_1, w_2, \ldots, w_n]$
  - Sometimes we call the weights $\theta = [\theta_1, \theta_2, \ldots, \theta_n]$
- Output: a predicted class $\hat{y}^{(i)} \in \{0,1\}$

multinomial logistic regression: $\hat{y}^{(i)} \in \{0,1, 2, 3, 4\}$

# How to do classification

- For each feature $x_i$, weight $w_i$ tells us importance of $x_i$
  - (Plus we'll have a bias $b$)
  - We'll sum up all the weighted features and the bias

$$z = \left(\sum_{i=1}^{n} w_i x_i\right) + b$$

$$z = w \cdot x + b$$

If this sum is high, we say $y=1$; if low, then $y=0$

# But we want a probabilistic classifier

We need to formalize “sum is high”

- We’d like a principled classifier that gives us a probability, just like Naive Bayes did
- We want a model that can tell us:
  - $p(y=1|x; \theta)$
  - $p(y=0|x; \theta)$

# The problem: z isn't a probability, it's just a number!

- z ranges from -∞ to ∞

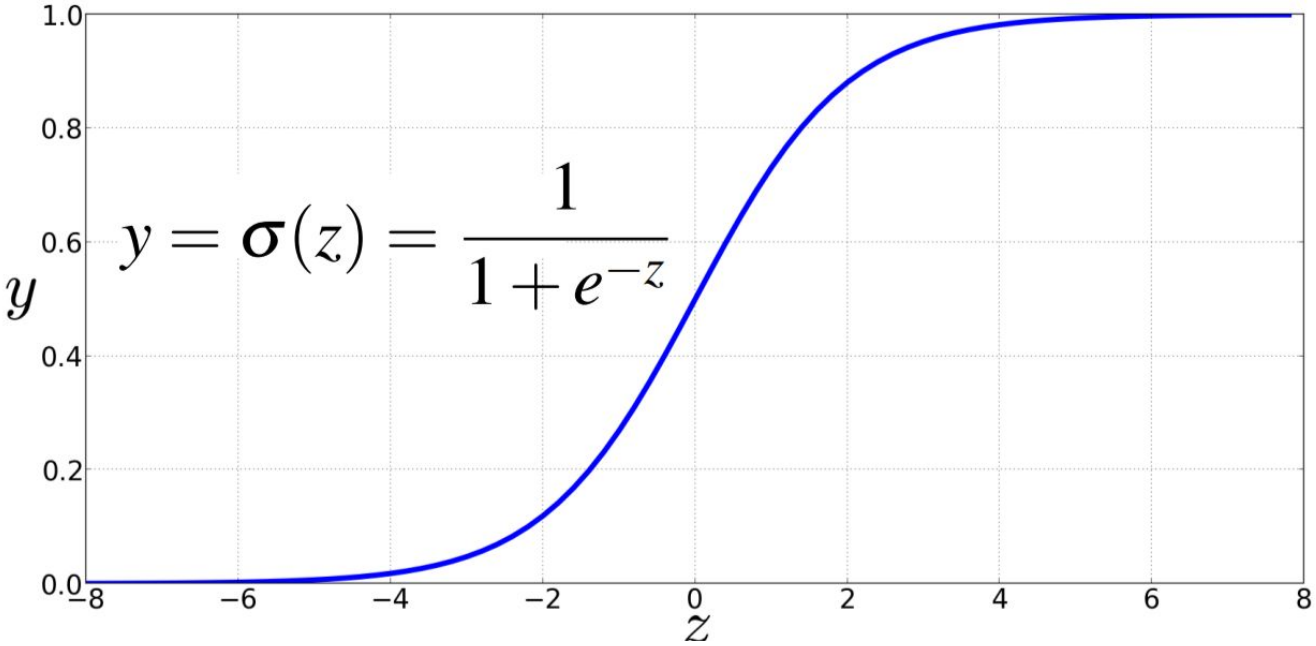
$$z = w \cdot x + b$$

- Solution: use a function of z that goes from 0 to 1

"sigmoid" or
"logistic" function

$$y = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-z)}$$

# The very useful sigmoid or logistic function



# Idea of logistic regression

- We'll compute w·x+b
- And then we'll pass it through the sigmoid function:

$$\sigma(w \cdot x + b)$$

- And we'll just treat it as a probability

# Making probabilities with sigmoids

$$
\begin{aligned}
P(y=1) &= \sigma(w \cdot x + b) \\
&= \frac{1}{1+\exp\left(-(w \cdot x + b)\right)}
\end{aligned}
$$

# Making probabilities with sigmoids

$$
\begin{aligned}
P(y=1) &= \sigma(w \cdot x + b) \\
&= \frac{1}{1+\exp\left(-(w \cdot x + b)\right)}
\end{aligned}
$$

$$
\begin{aligned}
P(y=0) &= 1 - \sigma(w \cdot x + b) \\
&= 1 - \frac{1}{1+\exp\left(-(w \cdot x + b)\right)} \\
&= \frac{\exp\left(-(w \cdot x + b)\right)}{1+\exp\left(-(w \cdot x + b)\right)}
\end{aligned}
$$

# By the way:

$$
\begin{aligned}
P(y=0) &= 1-\sigma(w\cdot x+b) \\
&= 1-\frac{1}{1+\exp(-(w\cdot x+b))} \\
&= \frac{\exp(-(w\cdot x+b))}{1+\exp(-(w\cdot x+b))}
\end{aligned}
\qquad = \sigma(-(w\cdot x+b))
$$

Because

$$1-\sigma(x)=\sigma(-x)$$

# Turning a probability into a classifier

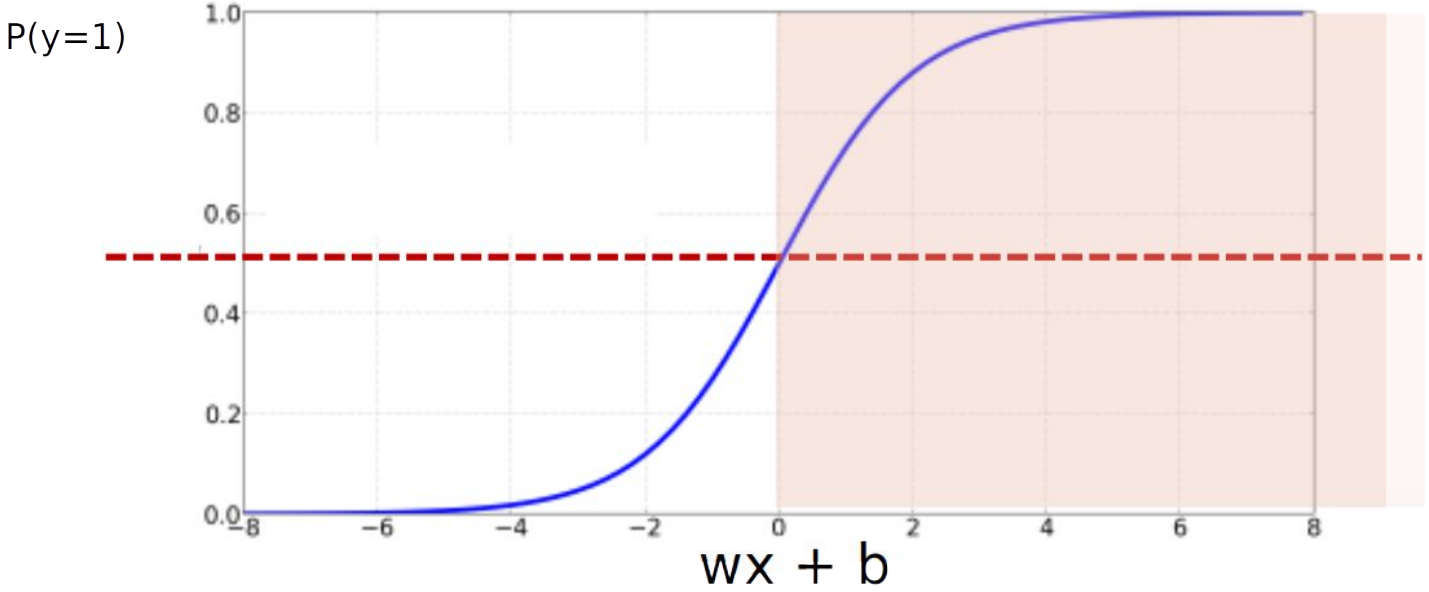
$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- 0.5 here is called the **decision boundary**

# The probabilistic classifier

$$P(y=1) = \sigma(w \cdot x + b)$$

$$= \frac{1}{1+\exp(-(w \cdot x + b))}$$



# Turning a probability into a classifier

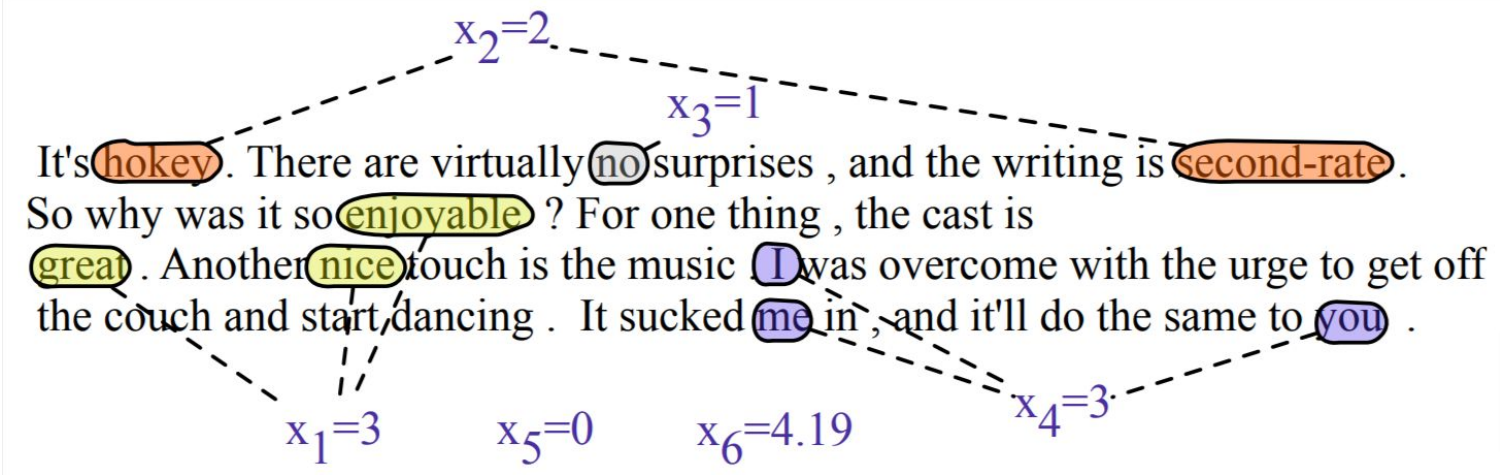
$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

if $w \cdot x + b > 0$

if $w \cdot x + b \leq 0$

# Sentiment example: does y=1 or y=0?

It's hokey . There are virtually no surprises , and the writing is second-rate . So why was it so enjoyable ? For one thing , the cast is great . Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .




| Var | Definition | Value |
|---|---|---|
| $x_1$ | count(positive lexicon) $\in$ doc) | 3 |
| $x_2$ | count(negative lexicon) $\in$ doc) | 2 |
| $x_3$ | $\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ | 1 |
| $x_4$ | count(1st and 2nd pronouns $\in$ doc) | 3 |
| $x_5$ | $\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ | 0 |
| $x_6$ | log(word count of doc) | $\ln(66) = 4.19$ |

# Classifying sentiment for input x

| Var | Definition | Value |
|---|---|---|
| $x_1$ | count(positive lexicon) $\in$ doc) | 3 |
| $x_2$ | count(negative lexicon) $\in$ doc) | 2 |
| $x_3$ | $\begin{cases} 1 & \text{if ``no'' } \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ | 1 |
| $x_4$ | count(1st and 2nd pronouns $\in$ doc) | 3 |
| $x_5$ | $\begin{cases} 1 & \text{if ``!'' } \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ | 0 |
| $x_6$ | log(word count of doc) | $\ln(66) = 4.19$ |

Suppose w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]

b = 0.1

# Classifying sentiment for input x

$$
\begin{aligned}
p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(.833) \\
&= 0.70
\end{aligned}
$$

$$
\begin{aligned}
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

# Scaling input features

- z-score

$$\mu_i = \frac{1}{m}\sum_{j=1}^{m} x_i^{(j)} \qquad \sigma_i = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\mathbf{x}_i^{(j)} - \mu_i\right)}$$

$$\mathbf{x}'_i = \frac{\mathbf{x}_i - \mu_i}{\sigma_i}$$

- normalize

$$\mathbf{x}'_i = \frac{\mathbf{x}_i - \min(\mathbf{x}_i)}{\max(\mathbf{x}_i) - \min(\mathbf{x}_i)}$$

# Wait, where did the W's come from?

- Supervised classification:
  - A training time we know the correct label $y$ (either $0$ or $1$) for each $x$.
  - But what the system produces at inference time is an estimate $\hat{y}$

# Wait, where did the W's come from?

- Supervised classification:
  - A training time we know the correct label $y$ (either $0$ or $1$) for each $x$.
  - But what the system produces at inference time is an estimate $\hat{y}$

- We want to set $w$ and $b$ to minimize the **distance** between our estimate $\hat{y}^{(i)}$ and the true $y^{(i)}$
  - We need a distance estimator: a **loss function** or a cost function
  - We need an **optimization algorithm** to update $w$ and $b$ to minimize the loss

# Learning components in LR

A loss function:

- **cross-entropy loss**

An optimization algorithm:

- **stochastic gradient descent**

# Loss function: the distance between ŷ and y

We want to know how far is the classifier output $\hat{y} = \sigma(w \cdot x + b)$

from the true output: y [= either 0 or 1]

We'll call this difference: $L(\hat{y},y)$ = how much $\hat{y}$ differs from the true y

# Intuition of negative log likelihood loss = cross-entropy loss

A case of conditional maximum likelihood estimation

We choose the parameters $w$,$b$ that maximize

- the log probability
- of the true $y$ labels in the training data
- given the observations $x$

# Next class:

- Deriving cross-entropy loss (please review Bernoulli distribution before class)
- Stochastic gradient descent
- Softmax