# Lecture 1

## Introduction to Natural Language Programming

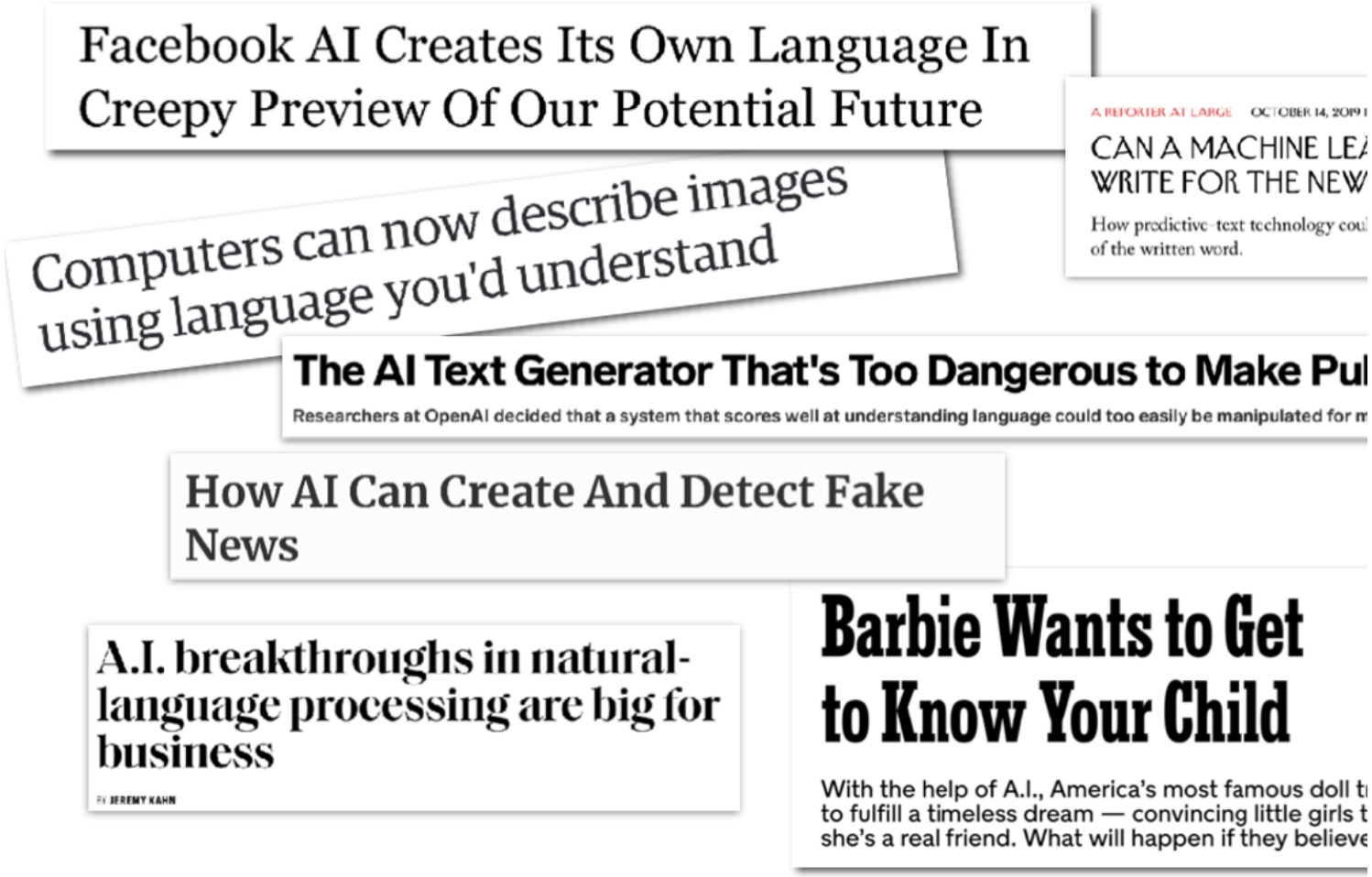
# What will you learn in this class?

- What is NLP?
  - The core tasks (as well as data sets and evaluation metrics) that people work on in NLP
- How does NLP work?
  - The fundamental models, algorithms and representations that have been developed for these tasks
- Why is NLP hard?
  - The relevant linguistic concepts and phenomena that must be handled to do well at these tasks

# The focus of this class

- We want to identify the structure and meaning of words, sentences, texts and conversations
- We mainly deal with language analysis/understanding, and less with language generation/production
- We focus on fundamental concepts, methods, models, and algorithms

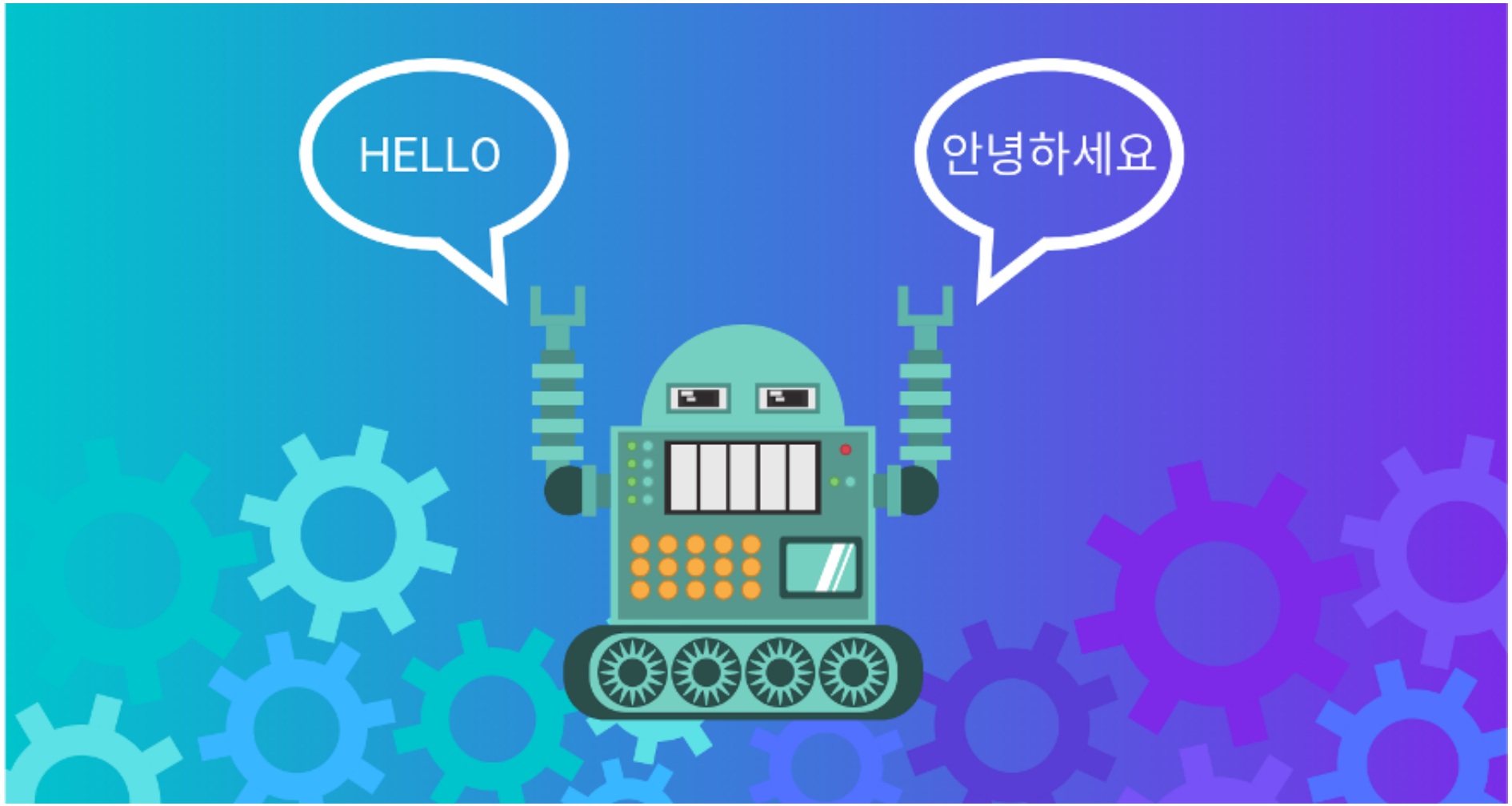
# Natural Language Processing is Everywhere!



# Dialog systems, chatbots, digital assistants

# IBM’s Watson wins at Jeopardy!

# Machine Translation



# What is the current state of NLP?

- Lots of commercial applications and interest.
  - Some applications are working pretty well already, others not so much
- A lot of hype around `deep learning` and `AI`
  - Neural nets are powerful classifiers and sequence models
  - Public libraries (Tensorflow, Pytorch, etc..) and datasets make it easy for anybody to get a model up and running
  - `End-to-end` models put into question whether we still need the traditional NLP pipeline that this class is built around
  - We're still in the middle of this paradigm shift
  - But many of the fundamental problems haven't gone away

# Huge language models solve NLP?

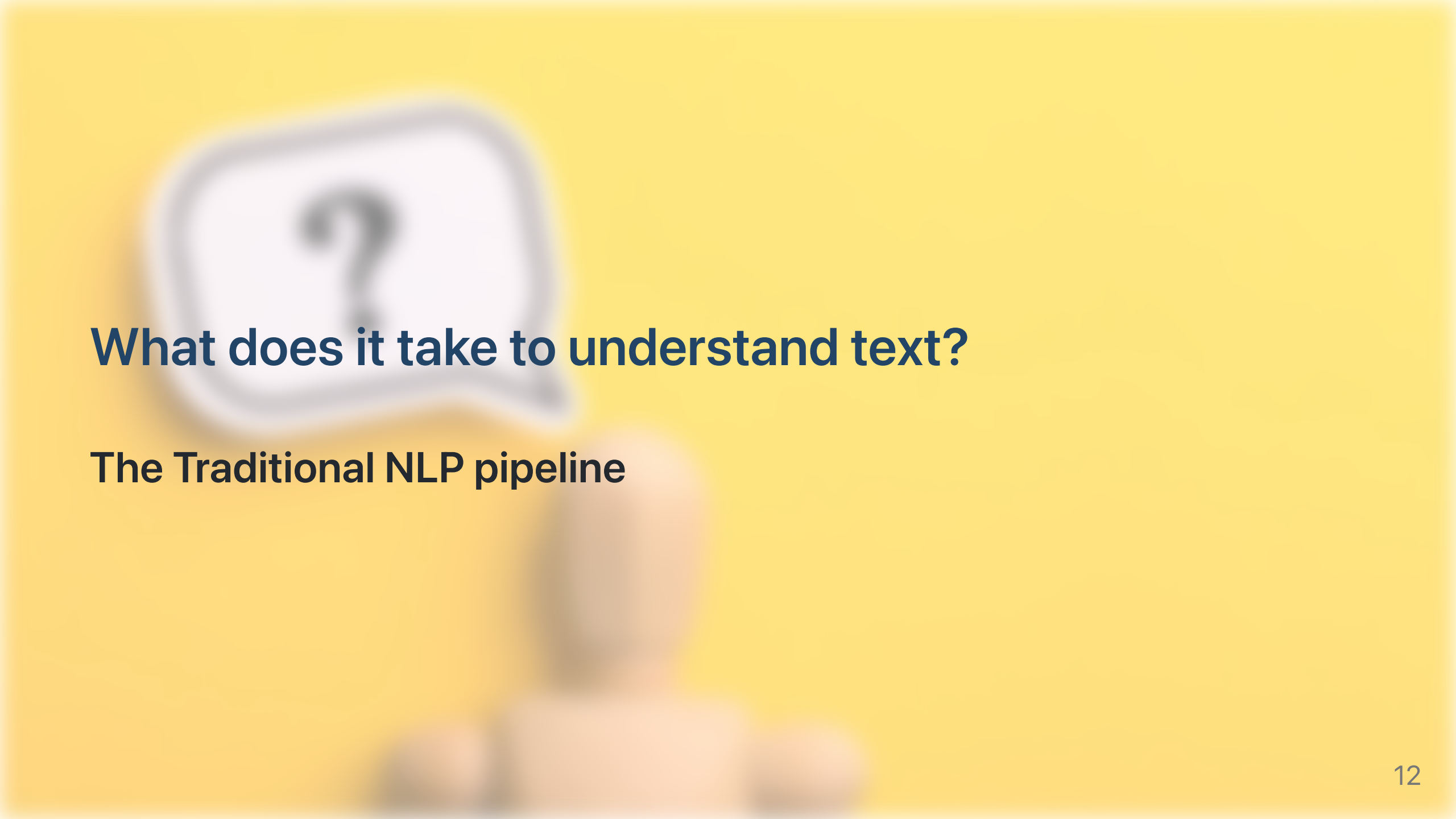
- A language model can be used to generate (produce) text
- Massive neural language models trained on vast amounts of text have been developed in the last few years
- Most recent incarnation: GPT-3 (175B parameters, trained on 300B tokens)
- But these models have no access to meaning.

# Examples of NLP applications (What can NLP be used for?)

- **Natural language (and speech) interfaces**
  - Search/IR, database access, image search, image description
  - Dialog systems (e.g., customer service, robots, cars, tutoring), chatbots
- **Information extraction, summarization, translation:**
  - Process (large amounts of) text automatically to obtain meaning/knowledge contained in the text
  - Identify/analyze trends, opinions, etc. (e.g., in social media)
  - Translate text automatically from one language to another
- **Convenience**:
  - Grammar/style checking, automate email filing, autograding

# Examples of NLP tasks (What capabilities do NLP systems need?)

- **Natural language understanding**
  - Extract information (e.g., about entities, events or relations between them) from text
  - Translate raw text into a meaning representation
  - Reason about information given in text
  - Execute NL instructions
- **Natural language generation and summarization**
  - Translate database entries or meaning representations to raw natural language text
  - Produce (appropriate) utterances/responses in a dialog
  - Summarize (newspaper or scientific) articles, describe images
- **Natural language translation**
  - Translate one natural language to another



# What does it take to understand text?

## The Traditional NLP pipeline

# Task: Tokenization/segmentation

- We need to split text into words and sentences.
  - Languages like Chinese or Thai don’t have spaces between words.
  - Even in English, this cannot be done deterministically:

```
There was an earthquake near D.C. You could even feel it in Philadelphia, New York, etc.
```

- NLP task:
  - What is the most likely segmentation/tokenization?

# Task: Part-of-speech-tagging

*Open the pod door, Hal.*

Verb Det Noun Noun , Name .

***Open the pod door , Hal .***

***open*:**
verb, adjective, or noun?
Verb: ***open*** *the door*
Adjective: *the* ***open*** *door*
Noun: *in the* ***open***

# How do we decide?

We want to know the most likely tags $T$
for the sentence $S$

$$\underset{T}{\operatorname{argmax}} P(T|S)$$

We need to define a statistical model of $P(T \mid S)$, e.g.:

$$\underset{T}{\operatorname{argmax}} P(T|S) = \underset{T}{\operatorname{argmax}} P(T)P(S|T)$$

$$P(T) =_{def} \prod_i P(t_i|t_{i-1})$$

$$P(S|T) =_{def} \prod_i P(w_i \mid t_i)$$

We need to estimate the parameters of $P(T|S)$, e.g.:

$$P(\, t_i = V \mid t_{i-1} = N\,) = 0.312$$

# Disambiguation requires statistical models

- Ambiguity is a core problem for any NLP task
- Statistical models are one of the main tools to deal with ambiguity.
- These models need to be trained (estimated, learned) before they can be used (tested, evaluated).

# "I made her duck"

What does this sentence mean?

"I made her crouch", "I cooked duck for her", "I cooked her [pet] duck (perhaps just for myself)", …

"***duck***": noun or verb?
"***make***": "*cook X*" or "*cause X to do Y*" ?
"***her***": "*for her*" or "*belonging to her*" ?

Language has different kinds of ambiguity, e.g.:

**Structural** ambiguity

"*I eat sushi* ***with tuna***" vs. "*I eat sushi* ***with chopsticks***"

"*I saw the man* ***with the telescope on the hill***"

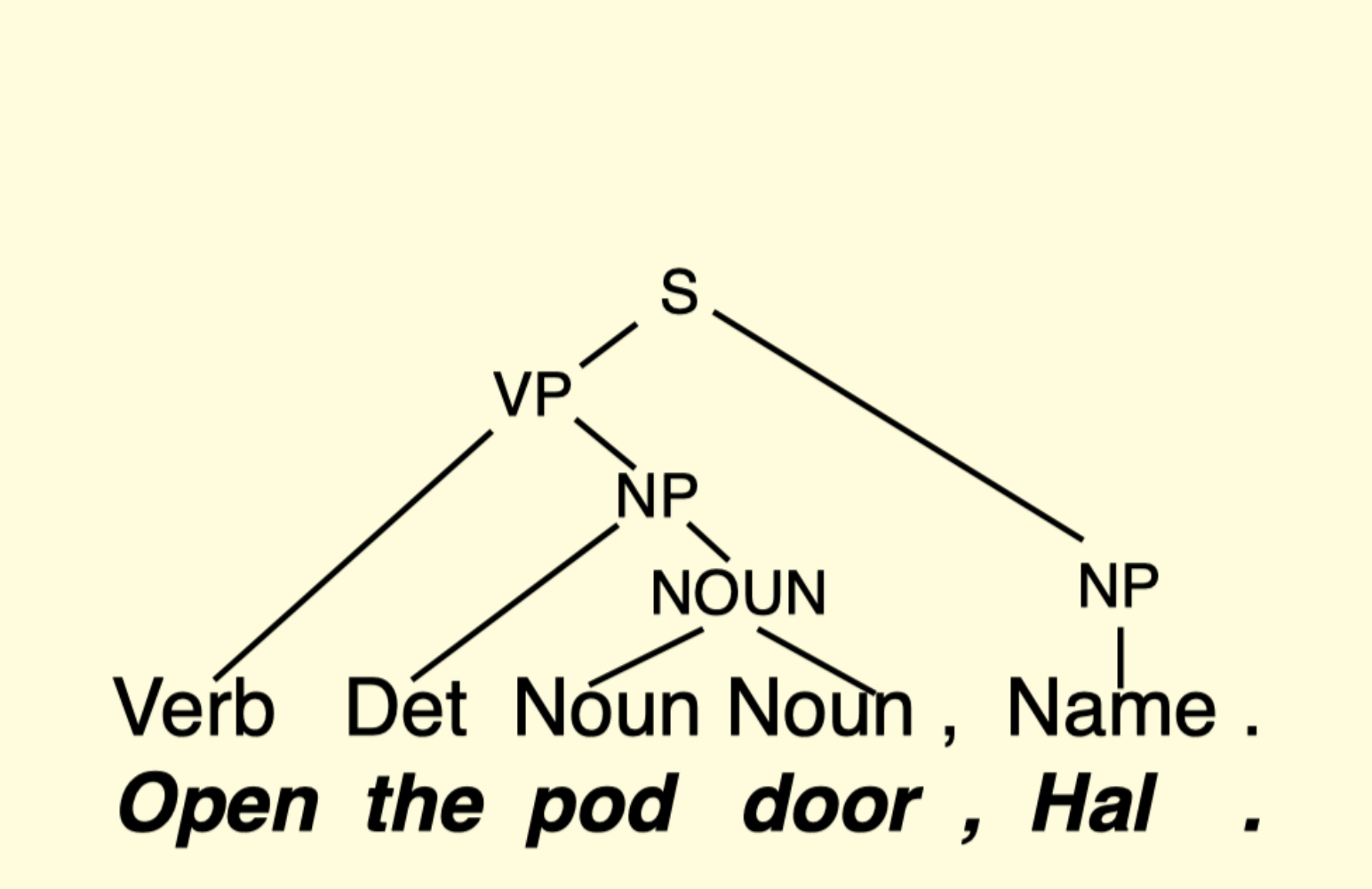
**Lexical (word sense)** ambiguity

"*I went to the* ***bank***": financial institution or river bank?

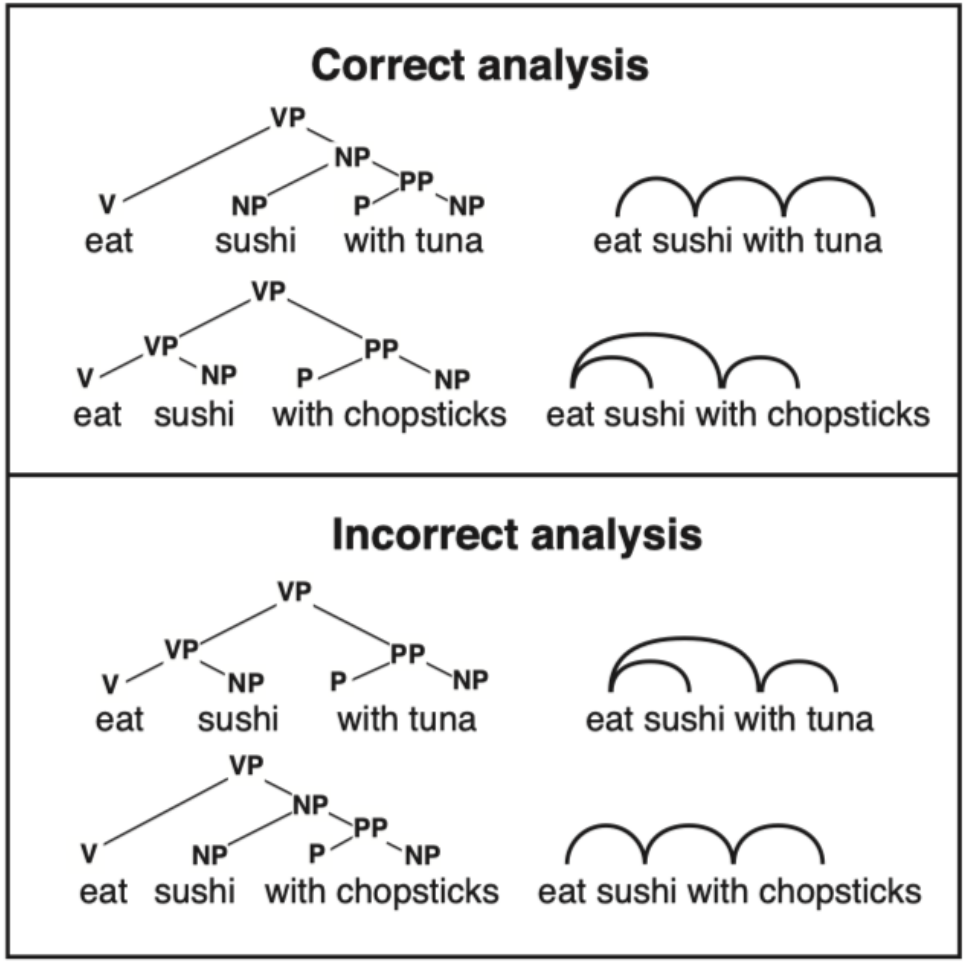
**Referential** ambiguity

"***John*** *saw* ***Jim***. ***He*** *was drinking coffee.*" Who was drinking coffee?

# Task: Syntactic parsing

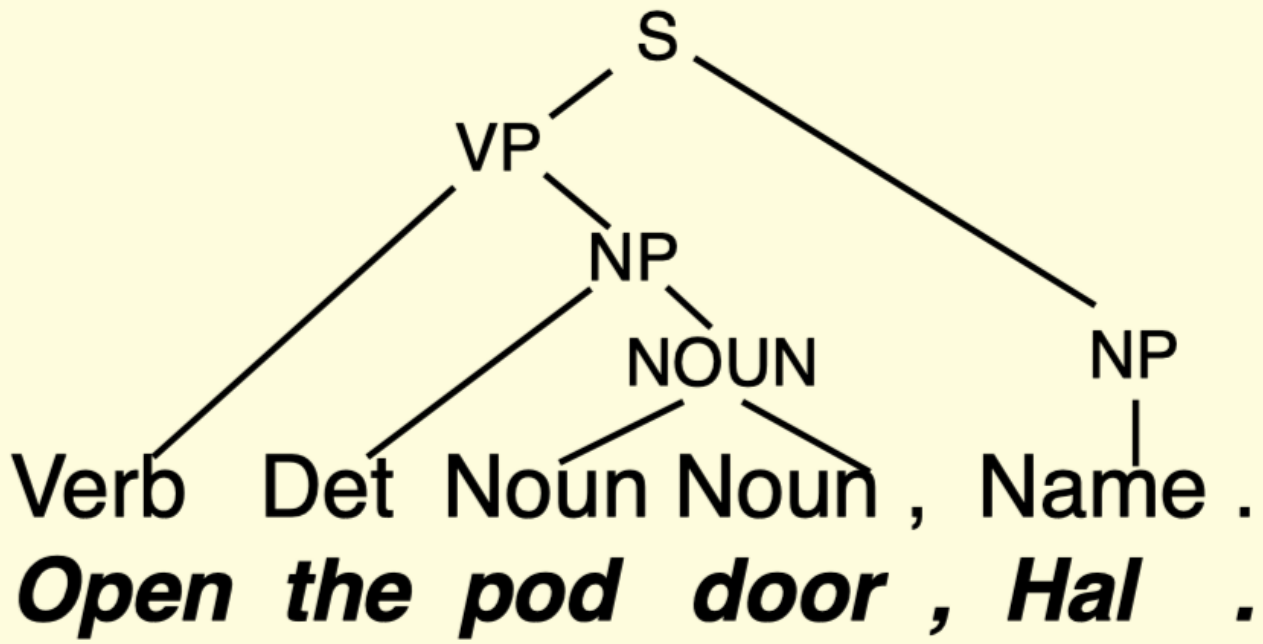


# Structure corresponds to meaning



# Task: Semantic analysis

$$\exists x \exists y(pod\_door(x) \ \& \ Hal(y) \ \& \ request(open(x, y)))$$



# Understanding texts

```
More than a decade ago, Carl Lewis stood on the threshold of what was
to become the greatest athletics career in history. He had just broken
two of the legendary Jesse Owens' college records, but never believed
he would become a corporate icon, the focus of hundreds of millions of
dollars in advertising. His sport was still nominally amateur.
Eighteen Olympic and World Championship gold medals and 21 world
records later, Lewis has become the richest man in the history of track
and field – a multi-millionaire.
```

- Who is Carl Lewis?
- Did Carl Lewis break any world records? (and how do you know that?)
- Is Carl Lewis wealthy? What about Jesse Owens?

# The NLP Pipeline

- **Tokenizer/Segmenter**
  to identify words and sentences
- **Morphological analyzer/POS-tagger**
  to identify the part of speech and structure of words
- **Word sense disambiguation**
  to identify the meaning of words
- **Syntactic/semantic Parser**
  to obtain the structure and meaning of sentences
- **Coreference resolution/discourse model**
  to keep track of the various entities and events mentioned

# NLP Pipeline: Assumptions

- Each step in the NLP pipeline embellishes the input with explicit information about its linguistic structure
  - POS tagging: parts of speech of word,
  - Syntactic parsing: grammatical structure of sentence,....
- Each step in the NLP pipeline requires its own explicit ( `symbolic` ) output representation:
  - POS tagging requires a POS tag set (e.g., NN=common noun singular, NNS = common noun plural, ...)
  - Syntactic parsing requires constituent or dependency labels (e.g., NP = noun phrase, or nsubj = nominal subject)
- These representations should capture linguistically appropriate generalizations/abstractions
  - Designing these representations requires linguistic expertise

# NLP Pipeline: Shortcomings

- Each step in the pipeline relies on a learned model that will return the most likely representations
  - This requires a lot of annotated training data for each step
  - Annotation is expensive and sometimes difficult (people are not 100% accurate)
  - These models are never 100% accurate
  - Models make more mistakes if their input contains mistakes
- How do we know that we have captured the `right` generalizations when designing representations?
  - Some representations are easier to predict than others
  - Some representations are more useful for the next steps in the pipeline than others
  - But we won't know how easy/useful a representation is until we have a model that we can plug into a particular pipeline

# Sidestepping the NLU pipeline

- Many current neural approaches for natural language understanding and generation go directly from the raw input to the desired final output.
- With large amounts of training data, this often works better than the traditional approach.

  -But these models don’t solve everything:
  - How do we incorporate knowledge, reasoning, etc. into these models?
  - What do we do when don’t have much training data?
    (e.g., when we work with a low-resource language)

# Why is NLP hard?

1. Ambiguity
2. Scale
3. Sparsity
4. Variation
5. Expressivity
6. Unmodeled variables
7. Unknown representation R

# Ambiguity

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

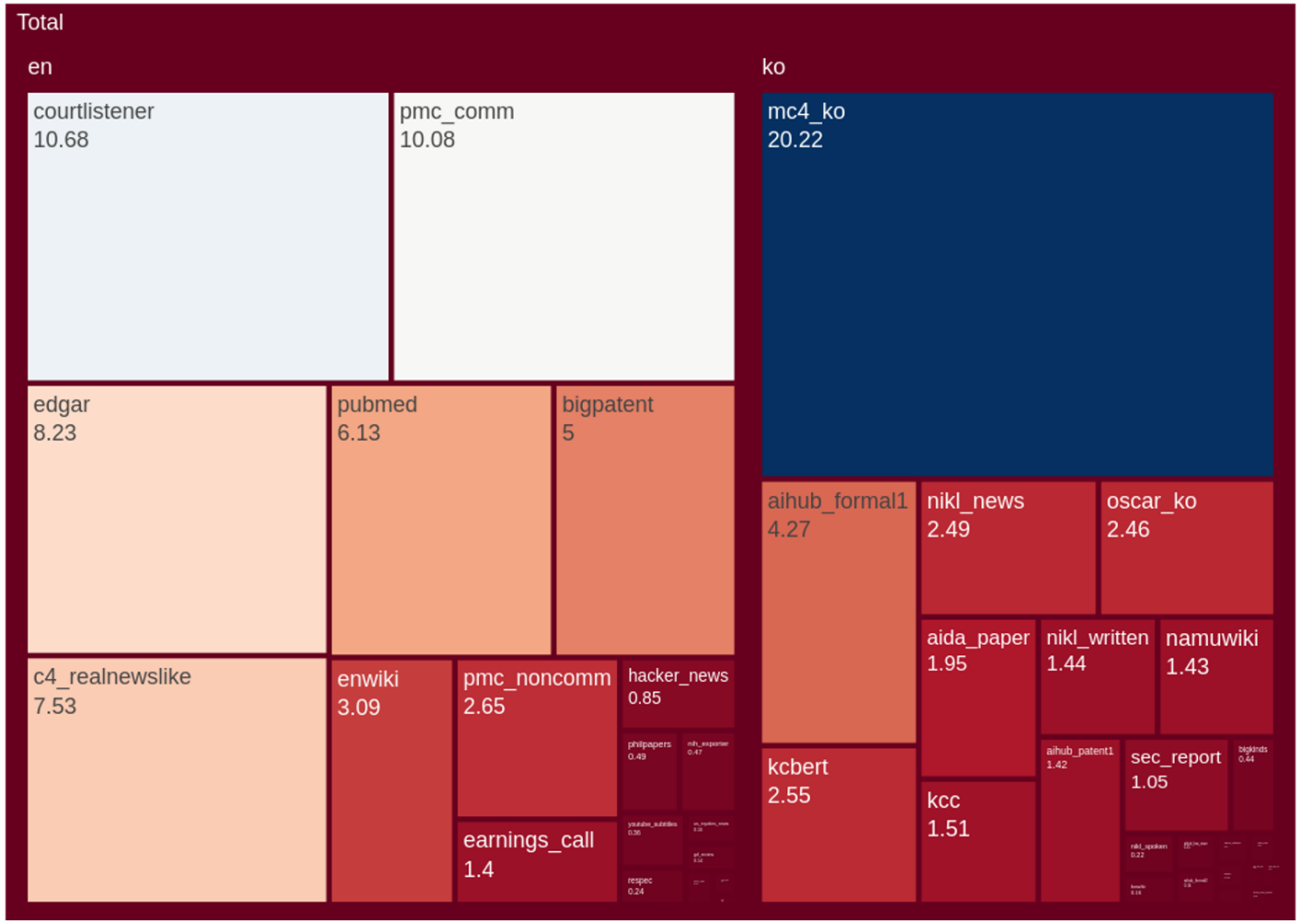
# Dealing with ambiguity

- How can we model ambiguity and choose correct analysis in context?
  - Non-probabilistic methods return all possible analyses.
  - Probabilistic models return best possible analysis,
    i.e., most probable one according to the model.
- But the “best” analysis is only good if our probabilities are accurate. Where do they come from?

# Corpora

- A corpus is a collection of text
  - Often annotated in some way
  - Sometimes just lots of raw text
- Examples
  - Penn Treebank: 1M words of parsed Wall Street Journal
  - Canadian Hansards: 10M+ words of aligned French/English sentences
  - Yelp reviews
  - The Web / Common Crawl: billions of words of who knows what

# The eKorpkit Corpus

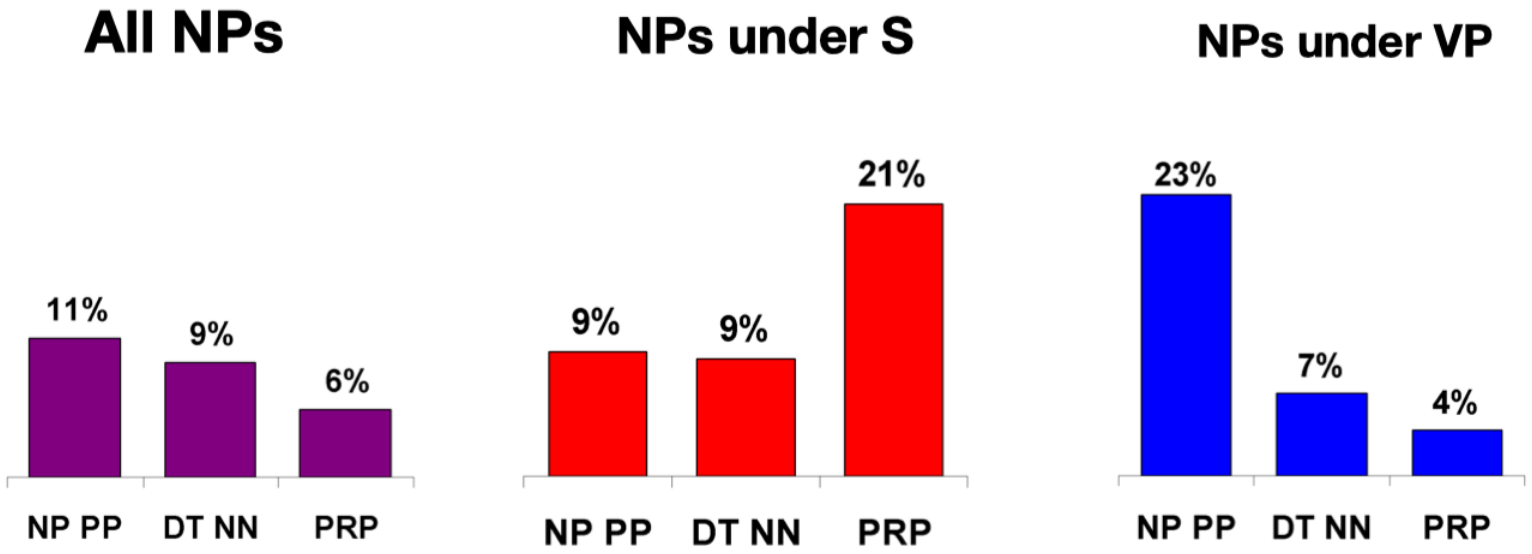



The eKorpkit Corpus is a large, diverse, multilingual (ko/en) language modelling dataset.

English: 258.83 GiB, Korean: 190.04 GiB, Total: 448.87 GiB

# Corpus-based methods

- Give us statistical information



# Statistical NLP

- Like most other parts of AI, NLP is dominated by statistical methods
  - Typically, more robust than earlier rule-based methods
  - Relevant statistics/probabilities learned from data
  - Normally requires lots of data about any particular phenomenon

# 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
- Zipf's Law
  - 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

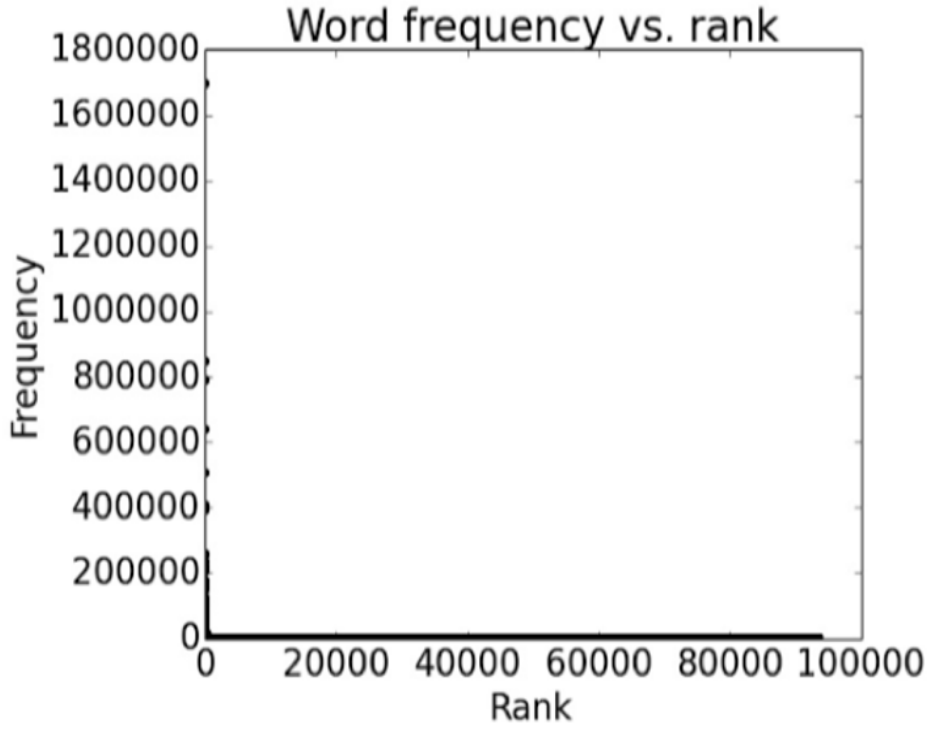
# Most frequent words in the English Europarl corpus (out of 24m word tokens)

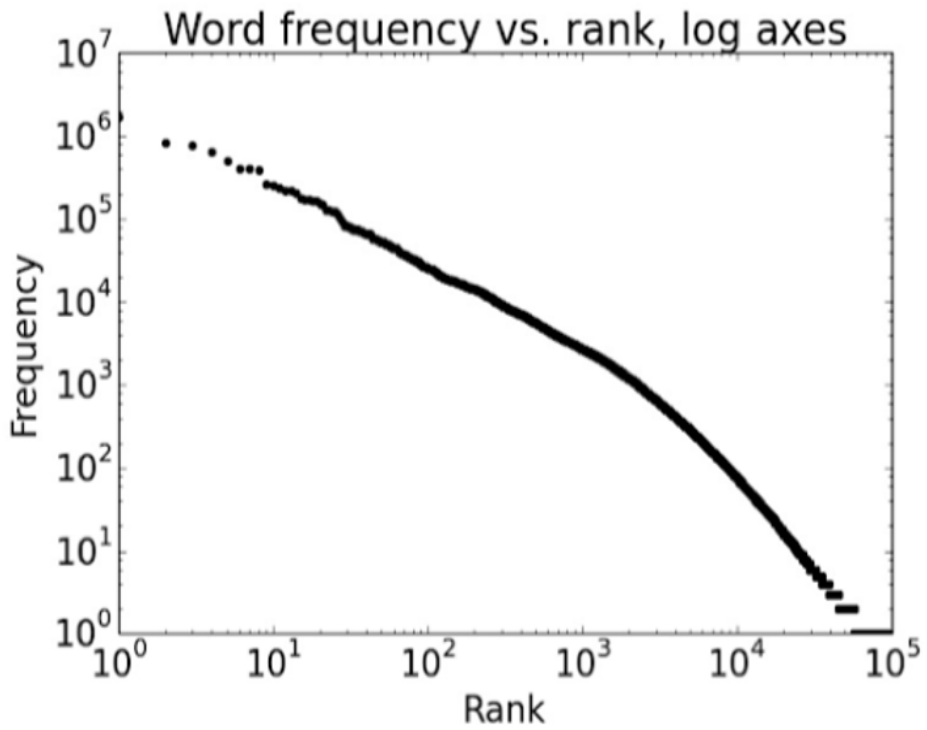
| any word | |
|---|---|
| Frequency | Token |
| 1,698,599 | the |
| 849,256 | of |
| 793,731 | to |
| 640,257 | and |
| 508,560 | in |
| 407,638 | that |
| 400,467 | is |
| 394,778 | a |
| 263,040 | I |

| nouns | |
|---|---|
| Frequency | Token |
| 124,598 | European |
| 104,325 | Mr |
| 92,195 | Commission |
| 66,781 | President |
| 62,867 | Parliament |
| 57,804 | Union |
| 53,683 | report |
| 53,547 | Council |
| 45,842 | States |

# Plotting word frequencies

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



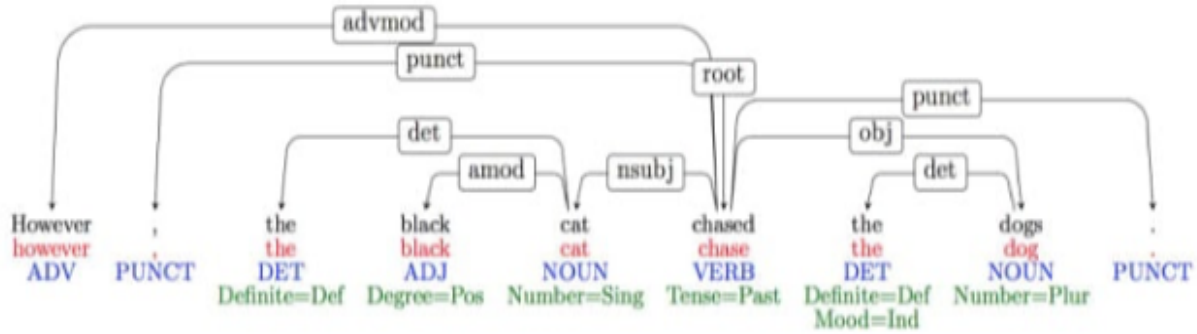



# Variation

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

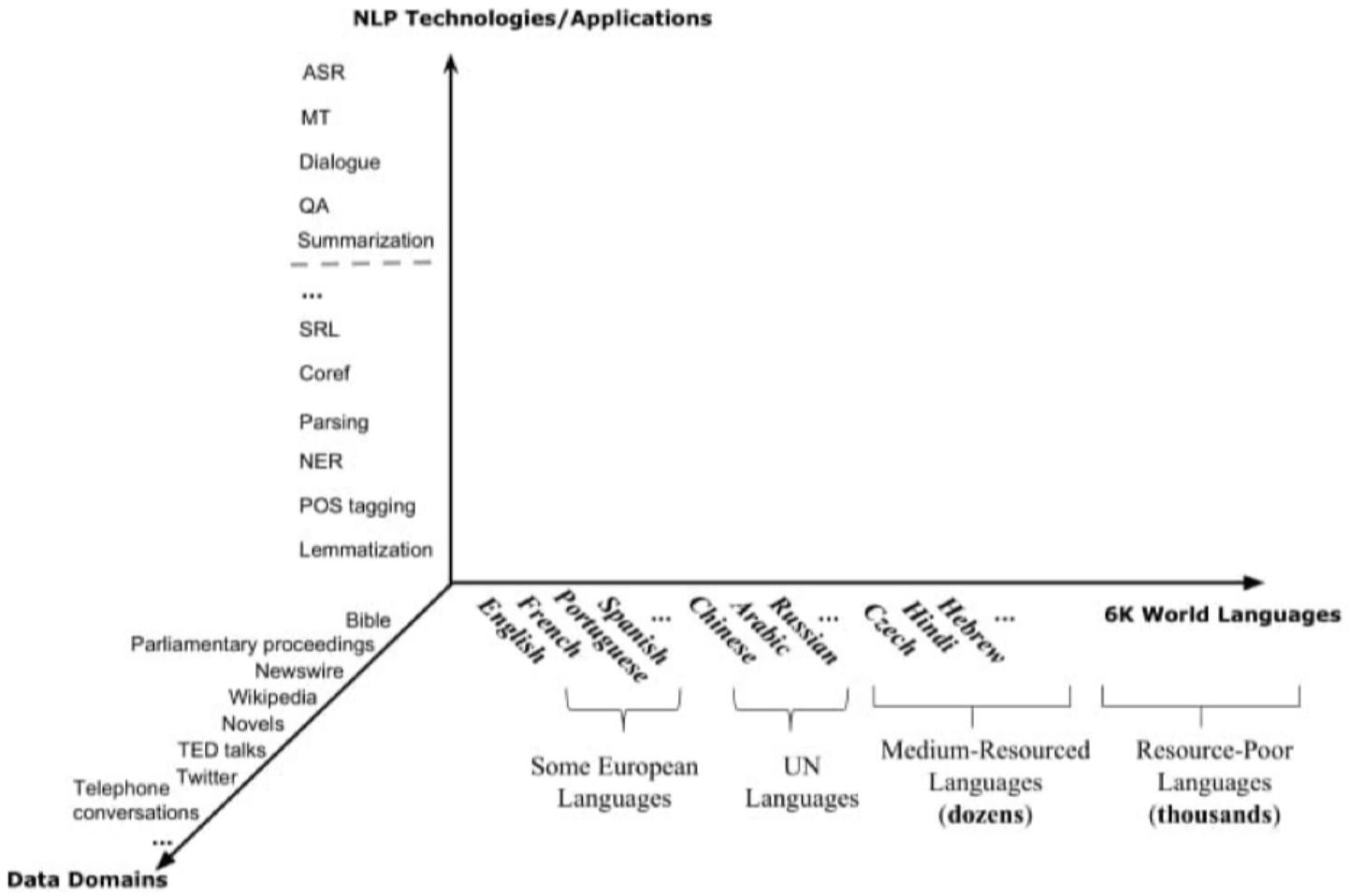

THE WALL STREET JOURNAL.

```
( (S (NP-SBJ (NP (NNP Pierre) (NNP Vinken))
             (, ,)
             (ADJP (NML (CD 61) (NNS years))
                   (JJ old))
             (, ,))
   (VP (MD will)
       (VP (VB join)
           (NP (DT the) (NN board))
           (PP-CLR (IN as)
                   (NP (DT a) (JJ nonexecutive) (NN director)))
           (NP-TMP (NNP Nov.) (CD 29)))))
   (. .)))
```




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


# Why is NLP Hard?



# Expressivity

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

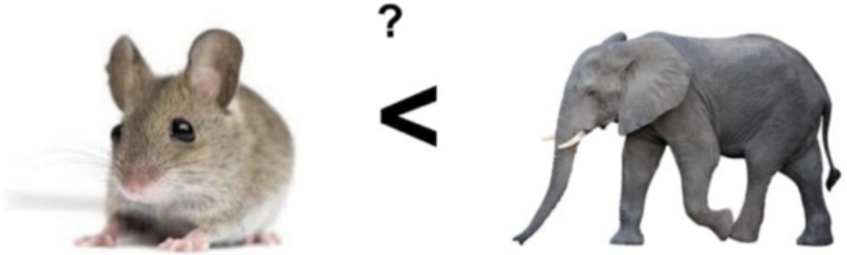
| | | |
|---|---|---|
| She gave the book to Deni. | vs. | She gave Deni the book. |
| Some kids popped by. | vs. | A few children visited. |
| Is that window still open? | vs. | Please close the window. |

# Unmodeled variables

- World knowledge
  - I dropped the glass on the floor, and it broke
  - I dropped the hammer on the glass, and it broke

“drink this milk.”

skater eats pavement



# Unknown representation

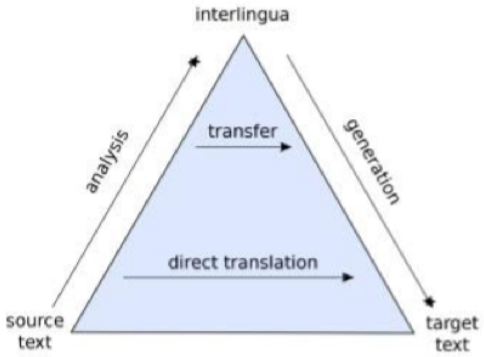
- Very difficult to capture what is $R$, since we don't even know how to represent the knowledge a human has/needs:
  - What is the "meaning" of a word, sentence, utterance?
  - How to model context?
  - Other general knowledge?

# Desiderata for NLP models

- Sensitivity to a wide range of phenomena and constraints in human language
- Generality across languages, modalities, genres, styles
- Strong formal guarantees (e.g., convergence, statistical efficiency, consistency)
- High accuracy when judged against expert annotations or test data
- Efficient
- Ethical

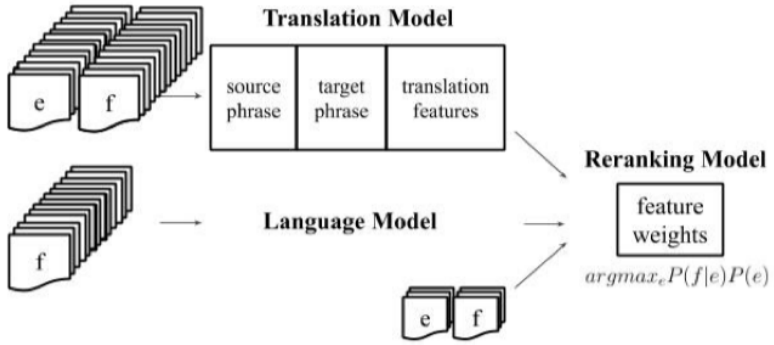
# Symbolic and probabilistic NLP

Logic/Rule-based NLP



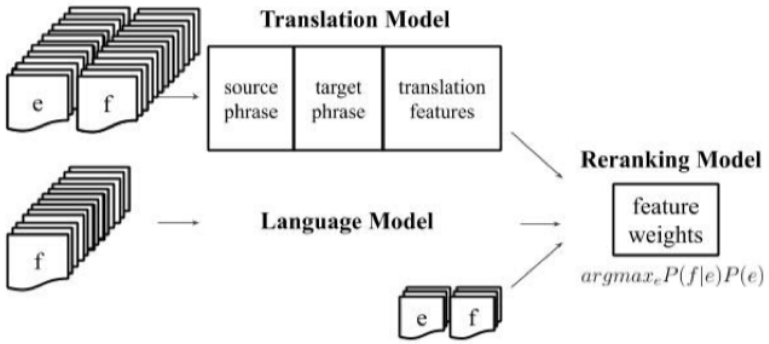

~1990s

Statistical NLP



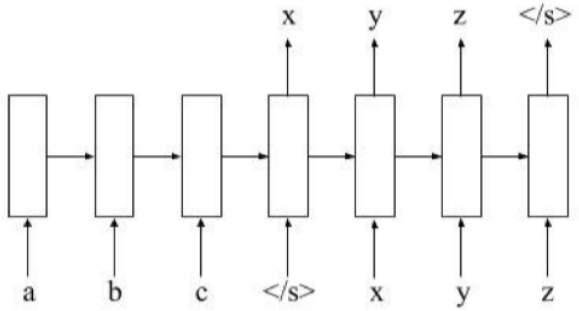
# Probabilistic and Deep NLP

Engineered features




~mid 2010s

Learned features



# Enjoy your class!