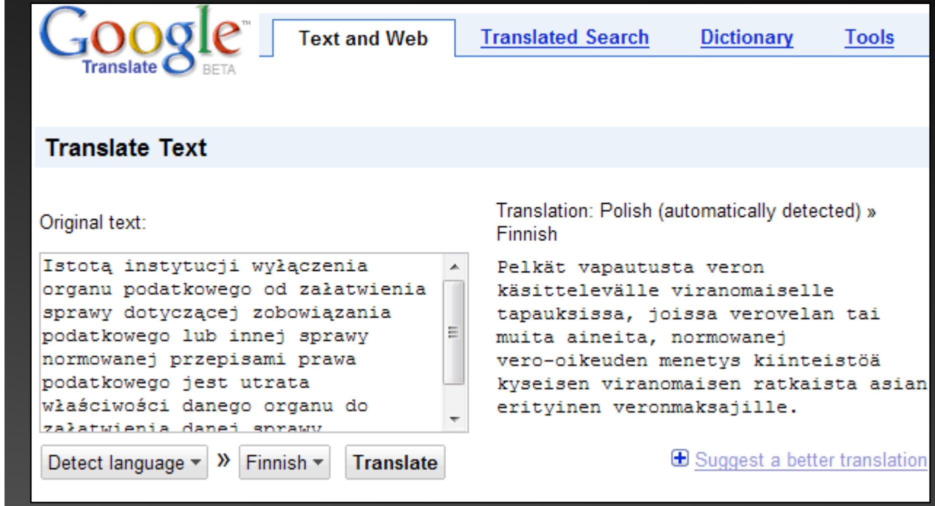
# Algorithms for Natural Language Processing
# Lecture 1
# Introduction

(today's slides based on those of Sharon Goldwater, Philipp Koehn, Alex Lascarides)

30 August 2017

# What is Natural Language Processing?



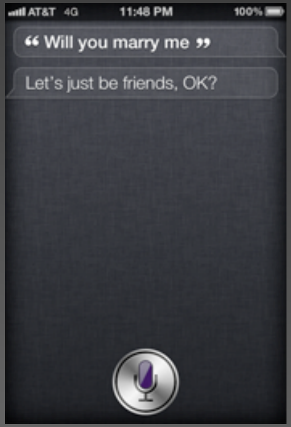
Translate Text

Original text:

Istotą instytucji wyłączenia organu podatkowego od załatwienia sprawy dotyczącej zobowiązania podatkowego lub innej sprawy normowanej przepisami prawa podatkowego jest utrata właściwości danego organu do załatwienia danej sprawy

Translation: Polish (automatically detected) » Finnish

Pelkät vapautusta veron käsittelevälle viranomaiselle tapauksissa, joissa verovelan tai muita aineita, normowanej vero-oikeuden menetys kiinteistöä kyseisen viranomaisen ratkaista asian erityinen veronmaksajille.



“ Will you marry me ”

Let's just be friends, OK?

# What is Natural Language Processing?

**Applications**

- Machine Translation
- Information Retrieval
- Question Answering
- Dialogue Systems
- Information Extraction
- Summarization
- Sentiment Analysis
- ...

**Core technologies**

- Language modelling
- Part-of-speech tagging
- Syntactic parsing
- Named-entity recognition
- Coreference resolution
- Word sense disambiguation
- Semantic Role Labelling
- ...

NLP lies at the intersection of **computational linguistics** and **artificial intelligence**. NLP is (to various degrees) informed by linguistics, but with practical/engineering rather than purely scientific aims. Processing **speech** (i.e., the acoustic signal) is separate.

# This course

NLP is a big field! We focus mainly on core ideas and methods needed for technologies in the second column (and eventually for applications).

- Linguistic facts and issues
- Computational models and algorithms, which
  - may involve expert specification of how to handle particular words and patterns in particular languages ("rule-based" or "symbolic")
  - may involve linguistic data + frequencies or machine learning to automatically determine generalization ("empirical" or "statistical")

# What are your goals?

Why are you here? Perhaps you want to:

- work at a company that uses NLP (perhaps as the sole language expert among engineers)
- use NLP tools for research in linguistics (or other domains where text data is important: social sciences, humanities, medicine, . . . )
- conduct research in NLP (or IR, MT, etc.)

# What does an NLP system need to “know”?

- Language consists of many levels of structure
- Humans fluently integrate all of these in producing/understanding language
- Ideally, so would a computer!

# Words

This is a simple sentence WORDS

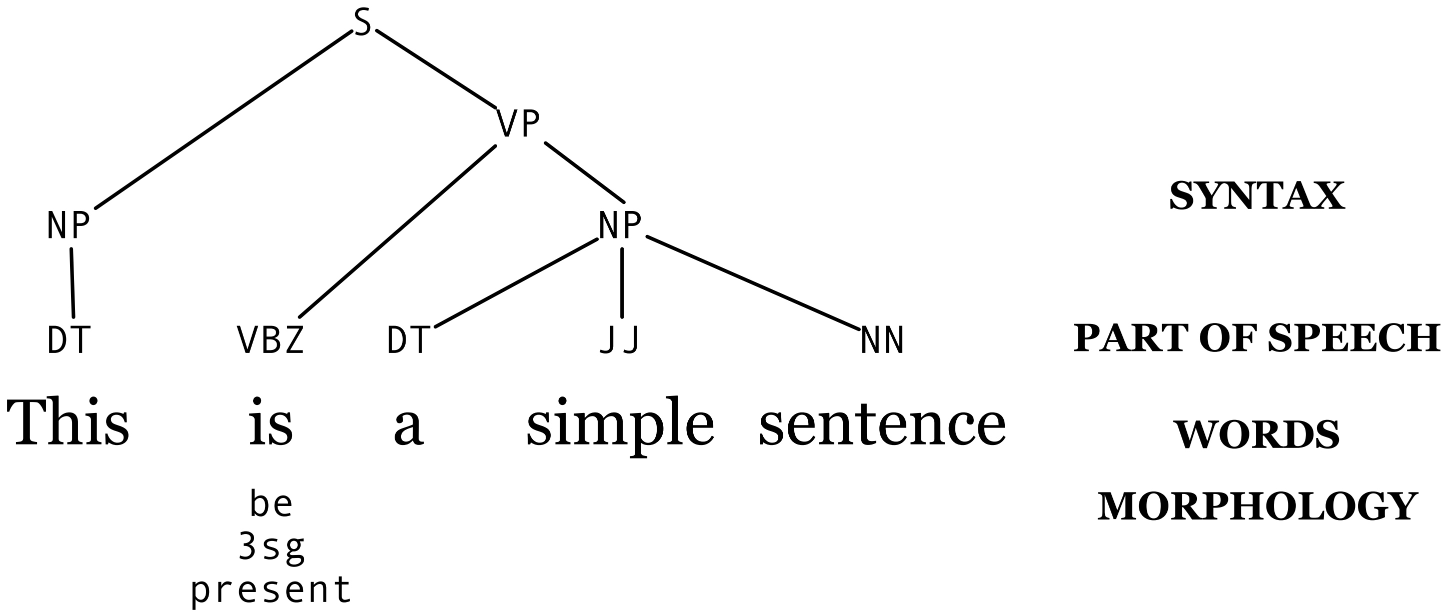
# Morphology

| | | | | | |
|---|---|---|---|---|---|
| This | is | a | simple | sentence | WORDS |
| | be<br>3sg<br>present | | | | MORPHOLOGY |

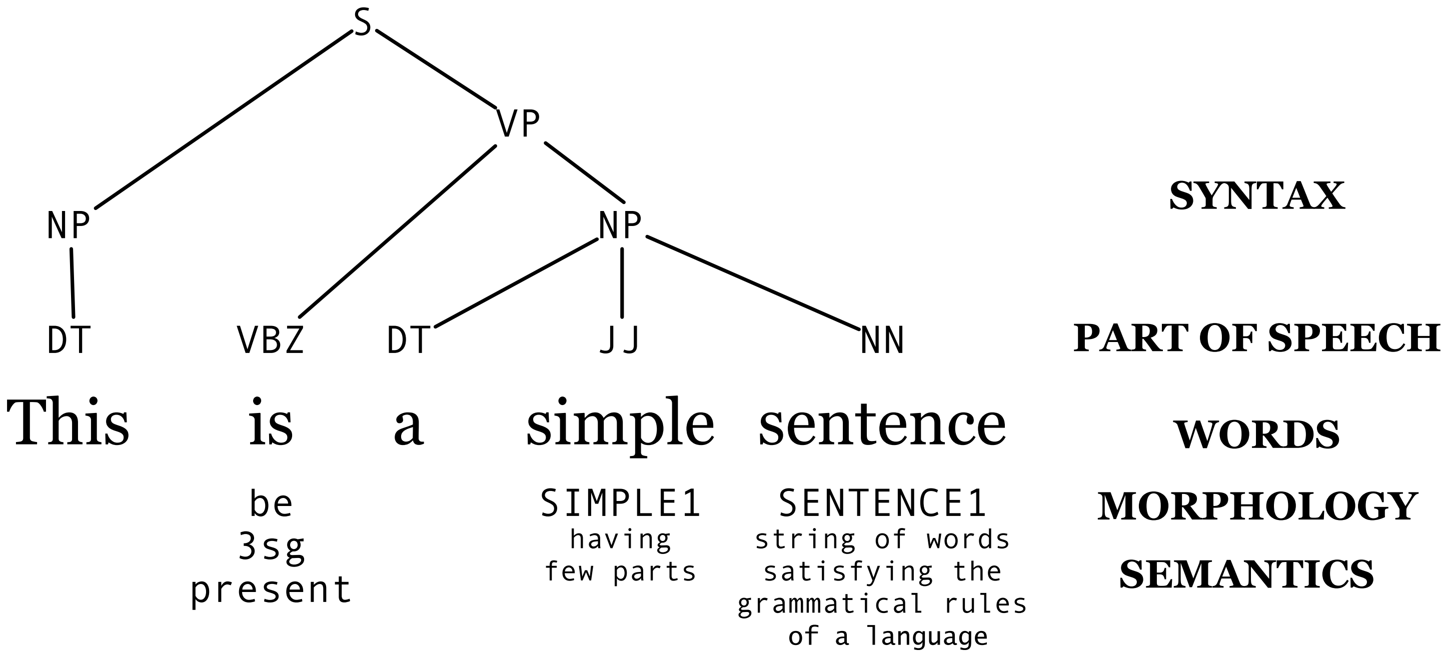
# Parts of Speech

| | | | | | |
|---|---|---|---|---|---|
| DT | VBZ | DT | JJ | NN | **PART OF SPEECH** |
| This | is | a | simple | sentence | **WORDS** |
| | be<br>3sg<br>present | | | | **MORPHOLOGY** |

# Syntax


S
VP
NP
NP
DT
VBZ
DT
JJ
NN
This
is
a
simple
sentence
be
3sg
present
SYNTAX
PART OF SPEECH
WORDS
MORPHOLOGY

# Semantics



S

VP

NP NP

DT VBZ DT JJ NN

This is a simple sentence

be
3sg
present

SIMPLE1
having
few parts

SENTENCE1
string of words
satisfying the
grammatical rules
of a language

SYNTAX

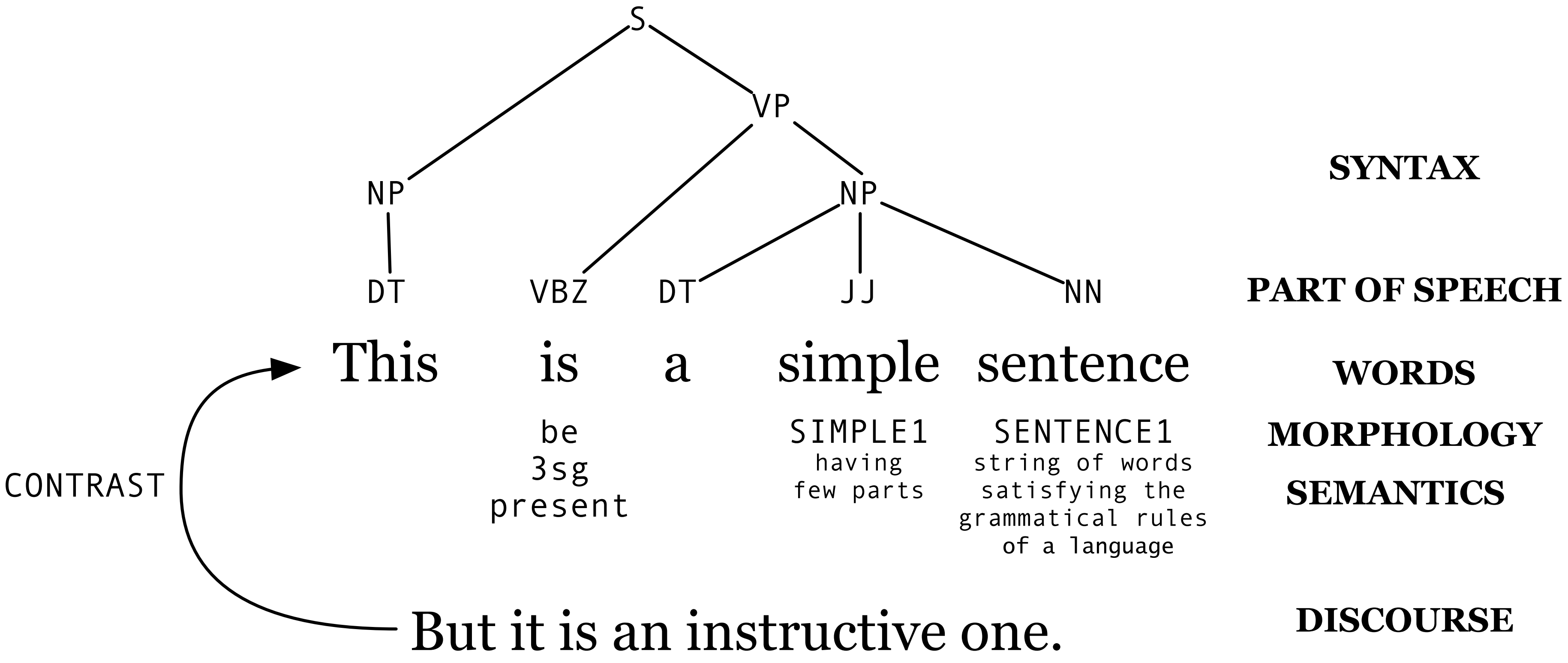
PART OF SPEECH

WORDS

MORPHOLOGY

SEMANTICS

# Discourse



S
VP
NP
NP
DT VBZ DT JJ NN

This is a simple sentence

be
3sg
present

SIMPLE1
having
few parts

SENTENCE1
string of words
satisfying the
grammatical rules
of a language

CONTRAST

But it is an instructive one.

**SYNTAX**

**PART OF SPEECH**

**WORDS**

**MORPHOLOGY**

**SEMANTICS**

**DISCOURSE**

# Why is NLP hard?

1. **Ambiguity** at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Quantifier scope: Every child loves some movie
- Multiple: I saw her duck

How can we model ambiguity, and choose the correct analysis in context?

# Ambiguity

What can we do about ambiguity?

- non-probabilistic methods (FSMs for morphology, CKY parsers for syntax) return *all possible analyses*.

- probabilistic models (HMMs for POS tagging, PCFGs for syntax) and algorithms (Viterbi, probabilistic CKY) return the *best possible analysis*.

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

# 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 are *learned from data*.
- Normally requires *lots of data* about any particular phenomenon.

# Why is NLP hard?

2. **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 (a great oversimplification, we'll return to this issue...)

# Word Counts

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 |

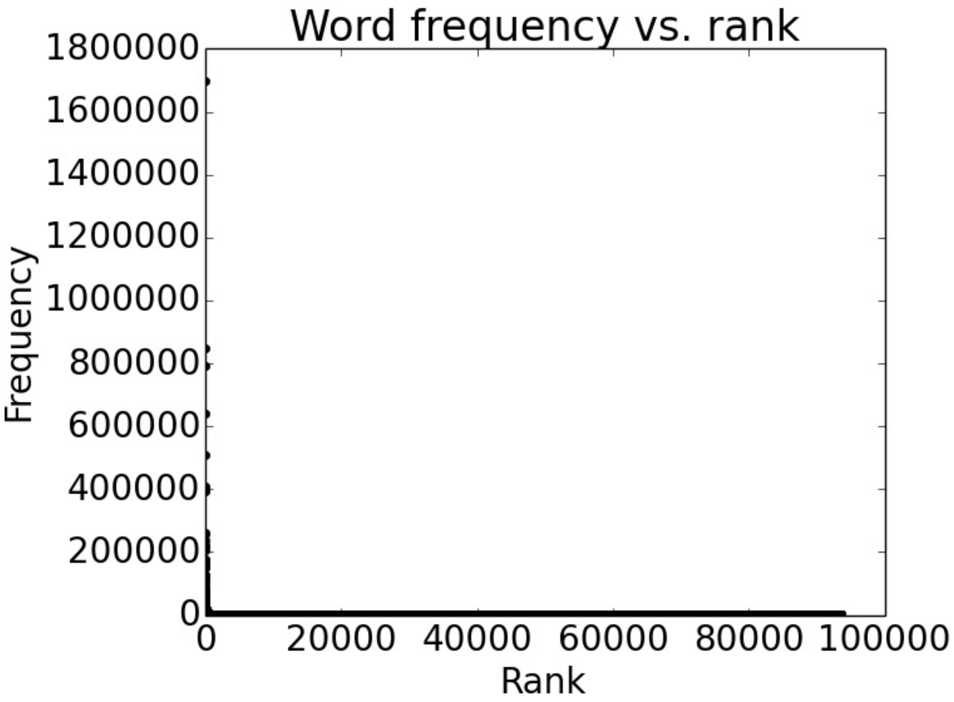
# Word Counts

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

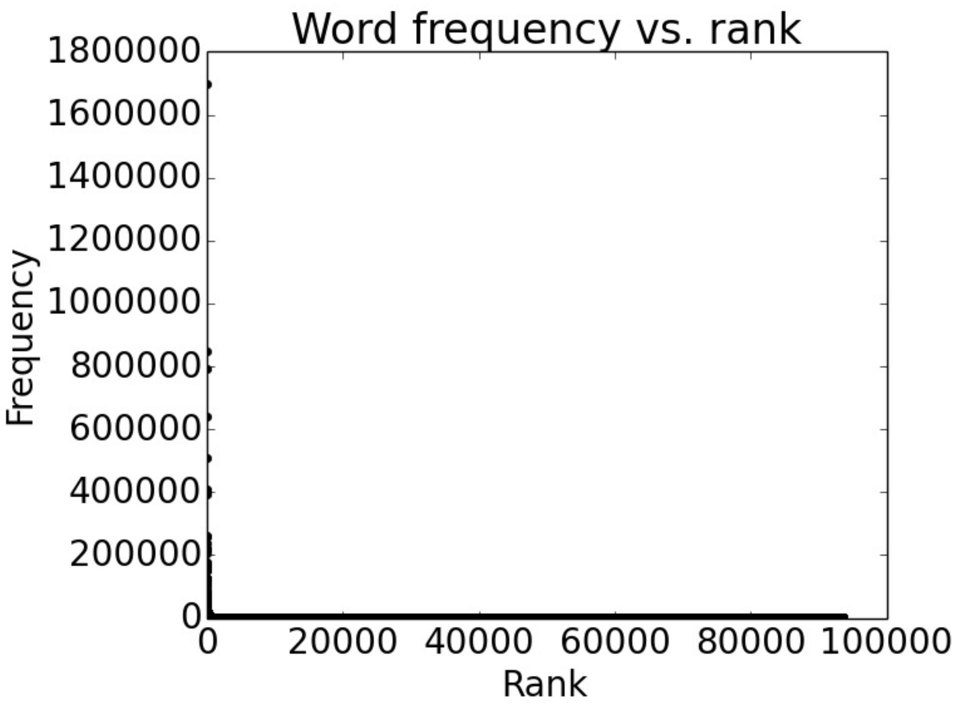
# Plotting word frequencies

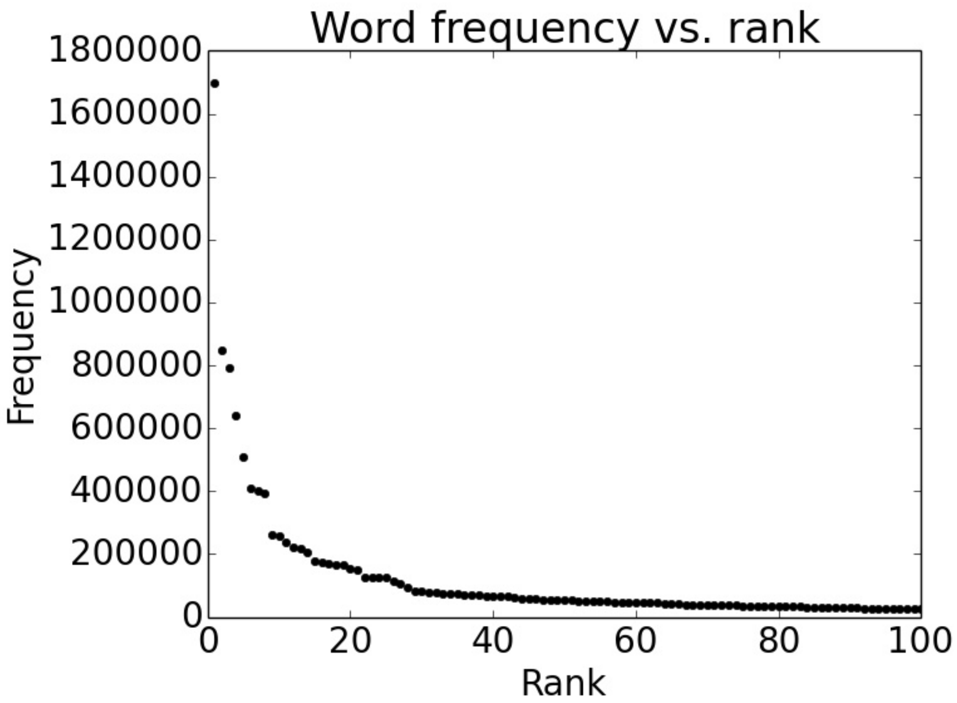
Order words by frequency. What is the frequency of $n$th ranked word?



# Plotting word frequencies

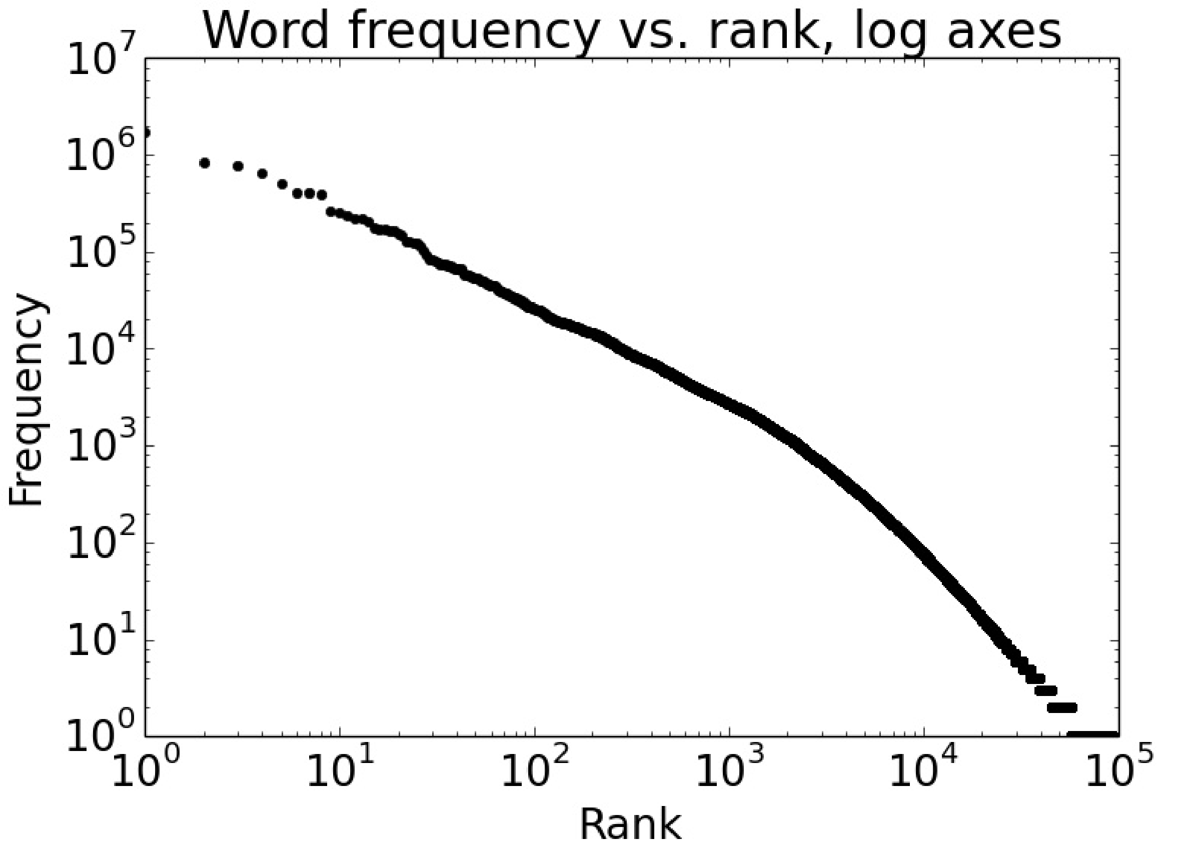
Order words by frequency. What is the frequency of $n$th ranked word?

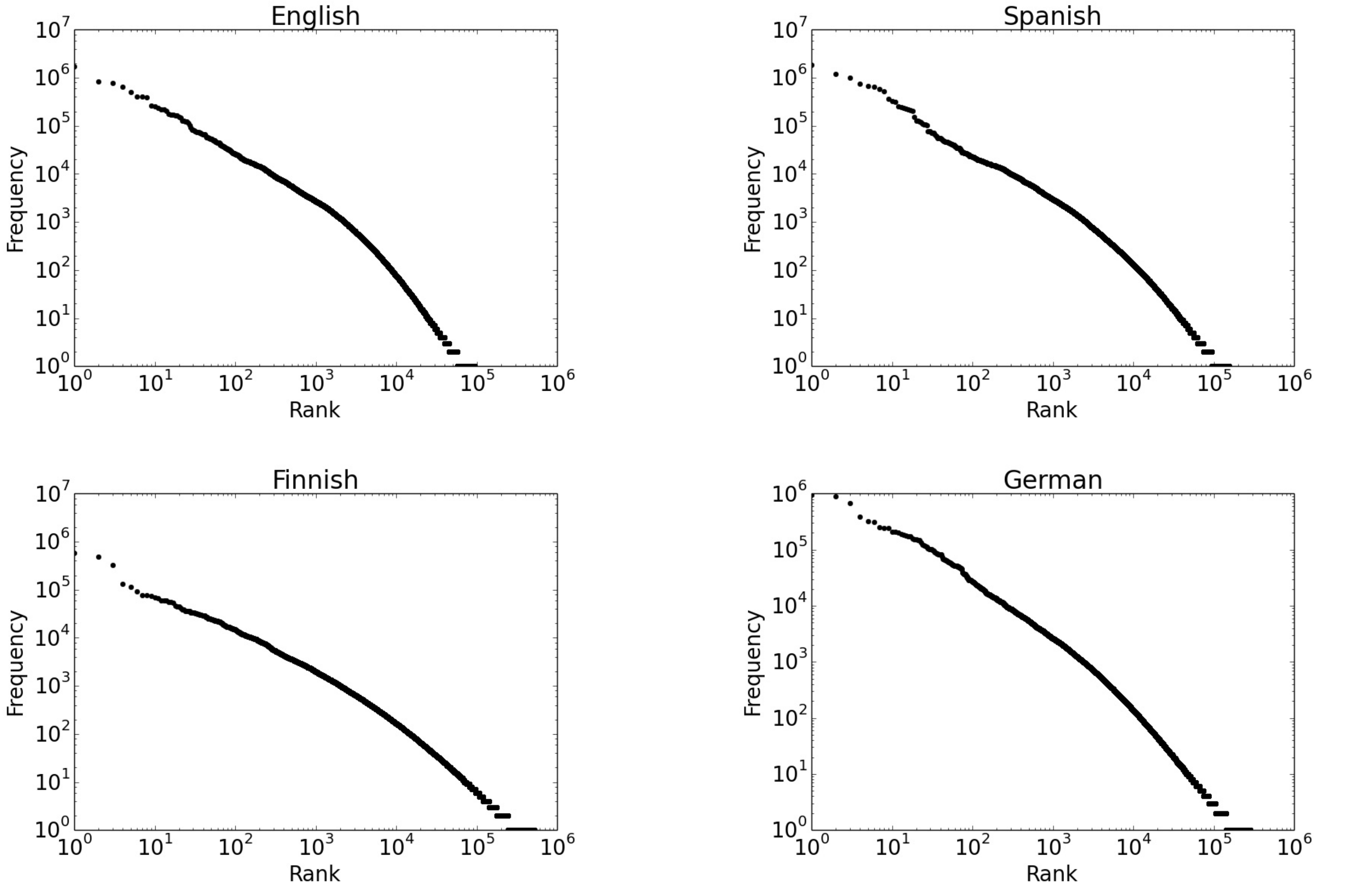




# Rescaling the axes

To really see what's going on, use logarithmic axes:




English
Spanish
Finnish
German
Frequency
Rank
$10^0$
$10^1$
$10^2$
$10^3$
$10^4$
$10^5$
$10^6$
$10^7$

# Zipf's law

Summarizes the behaviour we just saw:

$$f \times r \approx k$$

- $f$ = frequency of a word
- $r$ = rank of a word (if sorted by frequency)
- $k$ = a constant

# Zipf's law

Summarizes the behaviour we just saw:

$$f \times r \approx k$$

- $f =$ frequency of a word
- $r =$ rank of a word (if sorted by frequency)
- $k =$ a constant

Why a line in log-scales? $fr = k \quad \Rightarrow \quad f = \frac{k}{r} \quad \Rightarrow \quad \log f = \log k - \log r$

# Implications of Zipf's Law

- Regardless of how large our corpus is, there will be a lot of infrequent (and zero-frequency!) words.

- In fact, the same holds for many other levels of linguistic structure (e.g., syntactic rules in a CFG).

- This means we need to find clever ways to estimate probabilities for things we have rarely or never seen.

# Why is NLP hard?

3. **Variation**

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

> Mr./NNP Vinken/NNP is/VBZ chairman/NN of/IN Elsevier/NNP N.V./NNP ,/, the/DT Dutch/NNP publishing/VBG group/NN ./.

# Why is NLP hard?

3. **Variation**

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

Mr./NNP Vinken/NNP is/VBZ chairman/NN of/IN Elsevier/NNP N.V./NNP ,/, the/DT Dutch/NNP publishing/VBG group/NN ./.

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

ikr smh he asked fir yo last name

Twitter example due to Noah Smith

# Why is NLP hard?

4. **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 NLP hard?

5 and 6. **Context dependence** and **Unknown representation**

- Last example also shows that correct interpretation is context-dependent and often requires world knowledge.
- Very difficult to capture, 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?

# Themes

- Linguistic representations and resources for text
- Declarative models/specifications for procedural algorithms
- Balancing human expertise with machine pattern-finding algorithms
- Task definition and evaluation (quantitative, qualitative)
- Finite-state methods, probabilistic methods

# Organization of Topics (1/2)

Traditionally, NLP survey courses cover morphology, then syntax, then semantics and applications. This reflects the traditional form-focused orientation of the field, but this course will be organized differently, with the following units:

- **Introduction** (≈3 lectures): Getting everyone onto the same page with the fundamentals of text processing (in Python 3) and linguistics.

- **Words, Word formation, & Lexicons** (≈3 lectures): WordNet, a computational dictionary/thesaurus of word meanings; morphology and finite-state methods for composing or decomposing words

- **N-grams & Language Modeling** (≈3 lectures): Counting word sequences in data; probablistic models to predict the next word

- **Classification** (≈3 lectures): Assigning a category label to a document or word in context based on machine learning from data (supervised learning): naïve Bayes, perceptron

# Organization of Topics (2/2)

- **Sequence Tagging** (≈3 lectures): Probabilistic Hidden Markov Model (HMM) to assign labels (such as part-of-speech tags) to all the words in a sentence

- **Hierarchical Sentence Structure** (≈4 lectures): Analyzing the structure of how words in a sentence combine grammatically (syntax), and how their meanings combine (compositional semantics)

- **Distributional Semantics and Neural Networks** (≈2 lectures): Using large corpus evidence to compute vector representations of words, with and for neural networks

- **Applications** (≈3 lectures): Overviews of language technologies for text such as machine translation and question answering/dialog systems.

# Backgrounds (1/2)

This course has enrollment from two different majors!:

- Linguistics
- Computer Science

This means that there will be a diversity of backgrounds and skills, which is a fantastic opportunity for you to learn from fellow students. It also requires a bit of care to make sure the course is valuable for everyone.

# Backgrounds (2/2)

This course assumes that you are comfortable writing Python code without step-by-step guidance, and that you are familiar with discrete probability theory (e.g., conditional probability distributions).

Background in topics from formal language and automata theory, programming languages, machine learning, AI, logic, and linguistics will be useful but not assumed.

# What's *not* in this course

- Speech/signal processing, phonetics, phonology

(But see next slide!)

# Some Related Courses

- LING-362: Intro to NLP (Zeldes, this semester: also builds Python skills)
- LING-461: Signal Processing (Miller, this semester)
- LING-466: Machine Translation (Ruopp, next semester)
- LING-367: Computational Corpus Linguistics (Zeldes, this semester)
- COSC-270: Artificial Intelligence (Maloof, this semester)

# Course organization

- **Instructor:** Nathan Schneider; Office hour: M 5:00-6:00, Poulton 254
- **TA:** Harry Eldridge; Office hour: W 1:30-2:30, St. Mary's basement where UIS used to be

- **Lectures:** MW 3:30–4:45, Reiss 283
- **Textbook:** PDF drafts of Jurafsky and Martin, *Speech and Language Processing*, 3rd ed.: `http://web.stanford.edu/~jurafsky/slp3/`
- **Website:** for syllabus, schedule (lecture slides/readings/assignments): Canvas, and `http://tiny.cc/algnlp`

# Assessments

- **Homework assignments** (25%): Every 1–2 weeks. Most involve Python programming to practice implementing and experimenting with the algorithms discussed in class. We may give you framework code, but you should be able to translate conceptual understanding of an algorithm into an implementation, with appropriate Python data structures (e.g., lists, dicts, queues). We will look at your code and run it on the cs-class server, so please use Anaconda for Python 3.5+.

- **Written exams** (40%): Midterm = 15%, Final = 25%

- **Team project** (25%): Collaboratively developing a nontrivial system using NLP techniques from the course. In-class presentation and written project report.

- **Participation** (10%): Asking questions in class, participating on the Canvas discussion board, giving a 5 min. presentation about a language, quizzes.

# Policies

- **Attendance:** What is covered in class is important, so you should attend almost every session. Email the instructor in advance if you will have to miss class.

- **Late policy:** 25% off for each day late.

- **Exceptional circumstances:** If an emergency or major life difficulty may prevent you from attending and completing work in a timely fashion, alert the instructor/TA ASAP.

- **Academic integrity:** We expect you to do your own work unless it is specifically assigned as a group assignment/project. If you use someone else's idea, software library, etc., it should be documented. If it is a group assignment, the contributions of each member should be documented. Suspected dishonesty will be reported to the Honor Council.

(More detail on syllabus.)