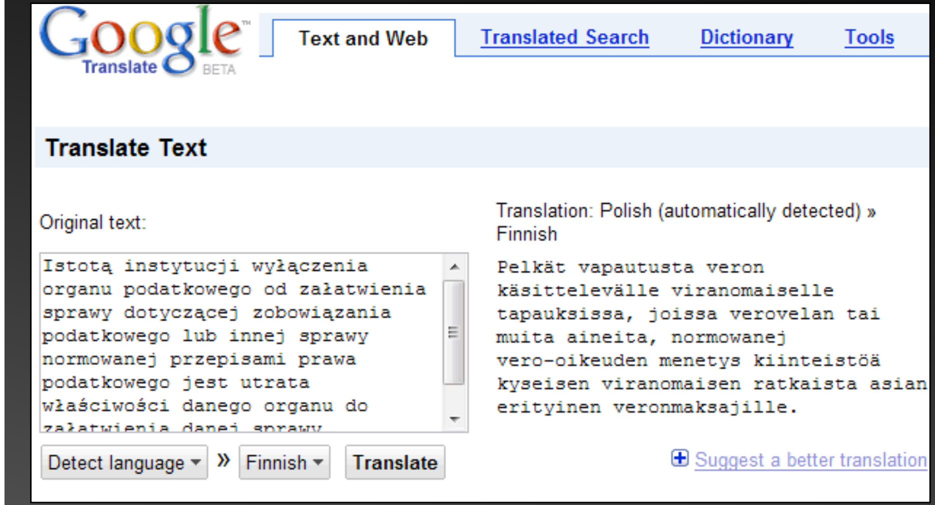
Empirical Methods in Natural Language Processing
Lecture 1
Introduction

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

9 January 2019

What is Natural Language Processing?



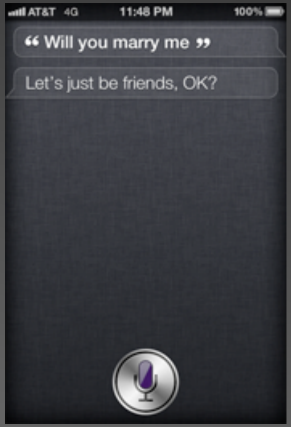
Translate Text

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, especially using data ("empirical")

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

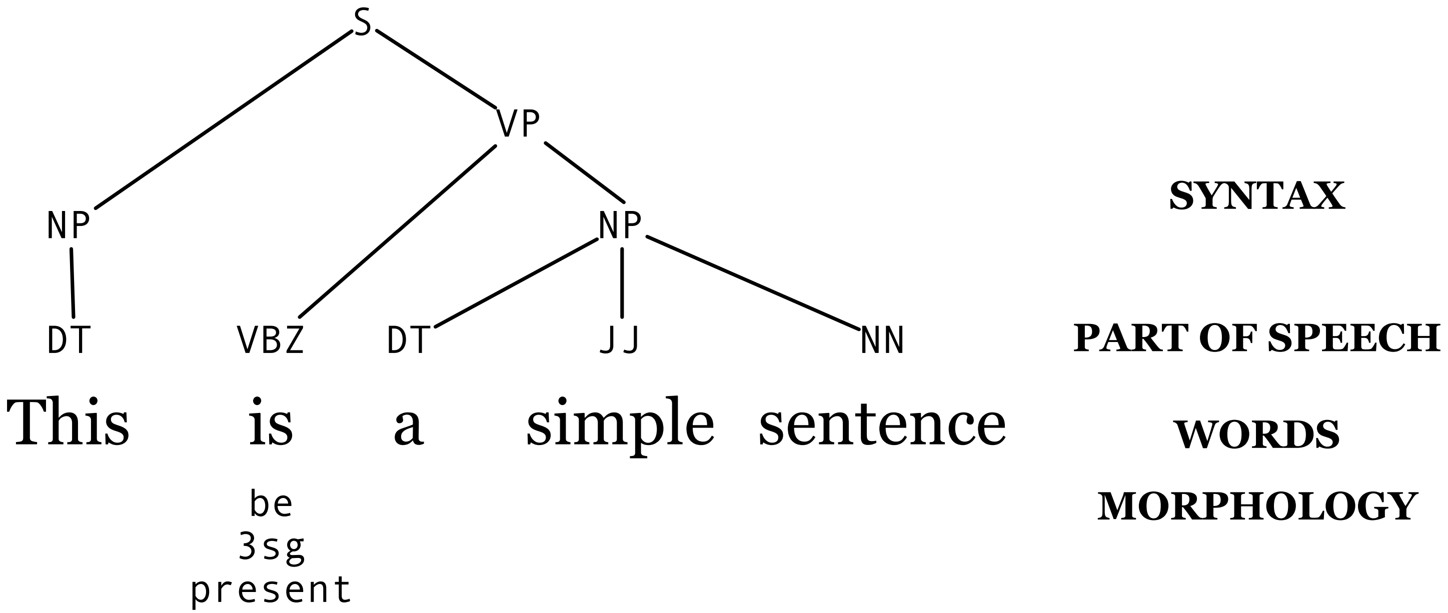
Morphology

This	is	a	simple	sentence	**WORDS**
	be 3sg present				**MORPHOLOGY**

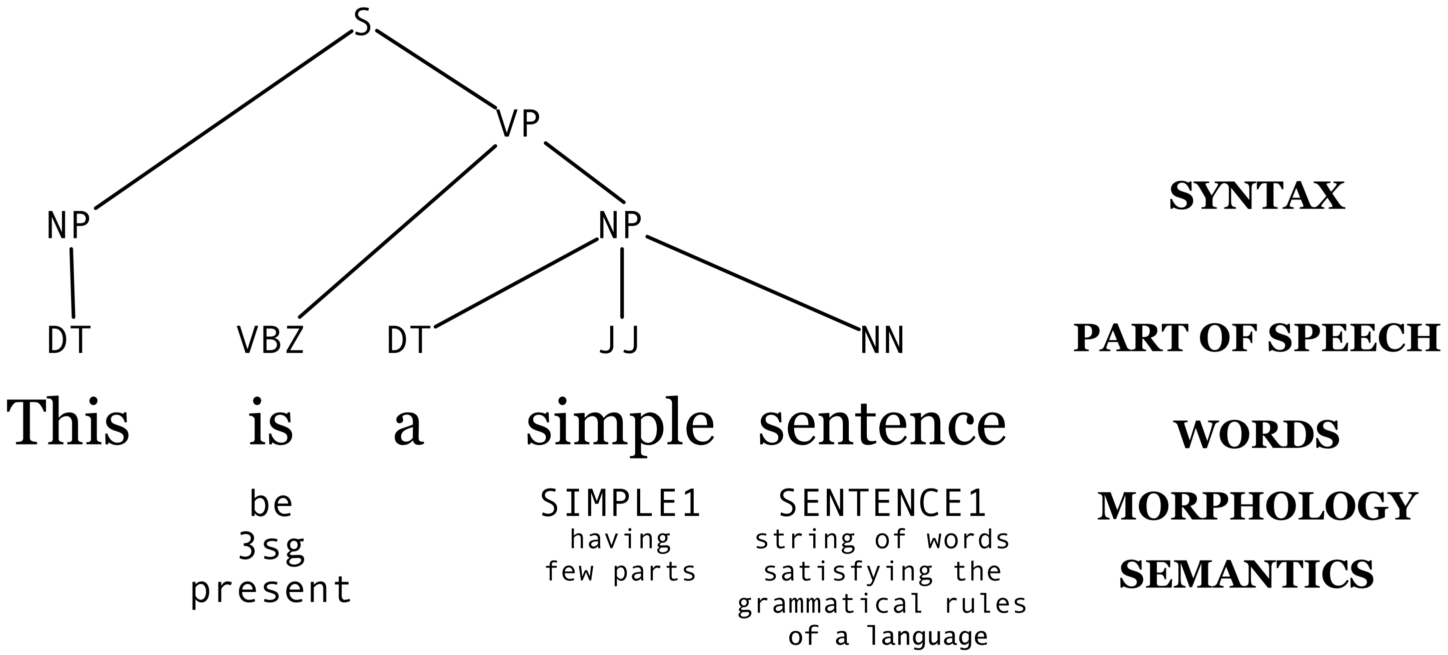
Parts of Speech

DT	VBZ	DT	JJ	NN	**PART OF SPEECH**
This	is	a	simple	sentence	**WORDS**
	be 3sg present				**MORPHOLOGY**

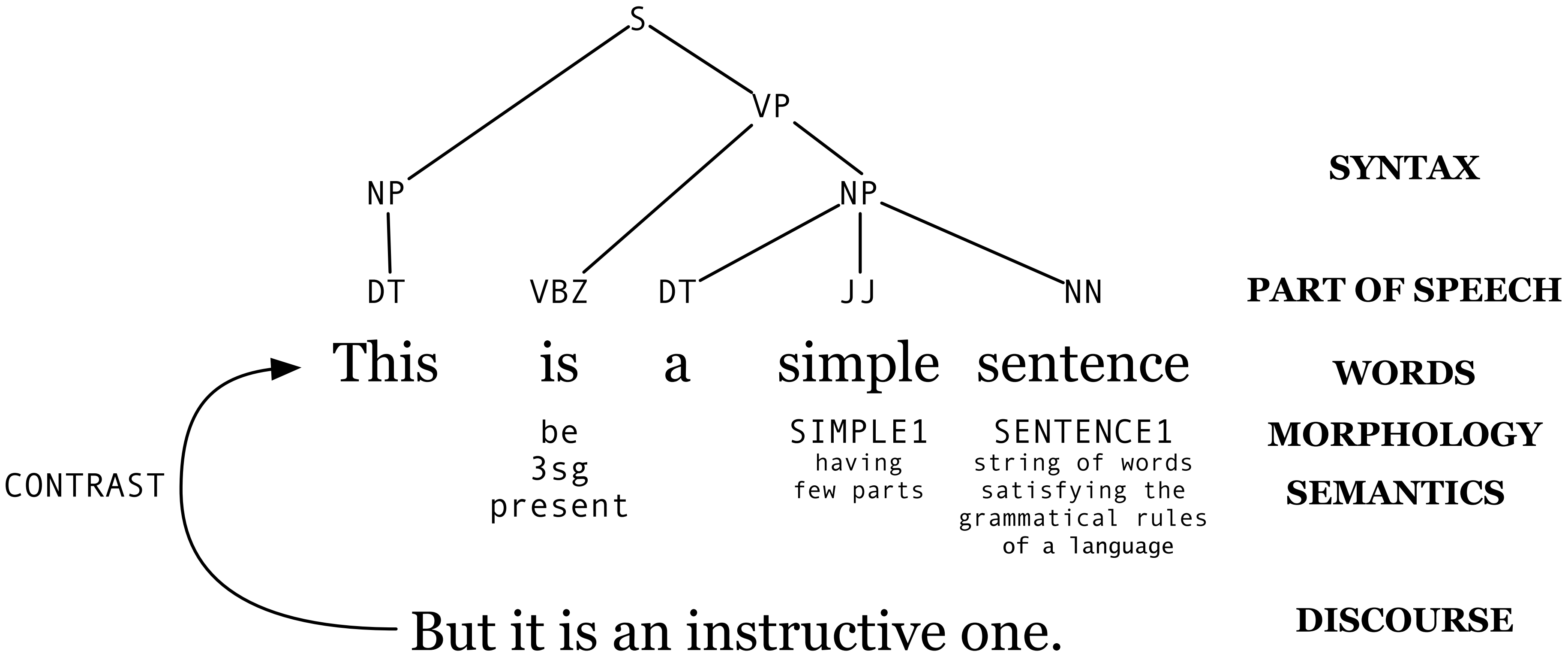
Syntax


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

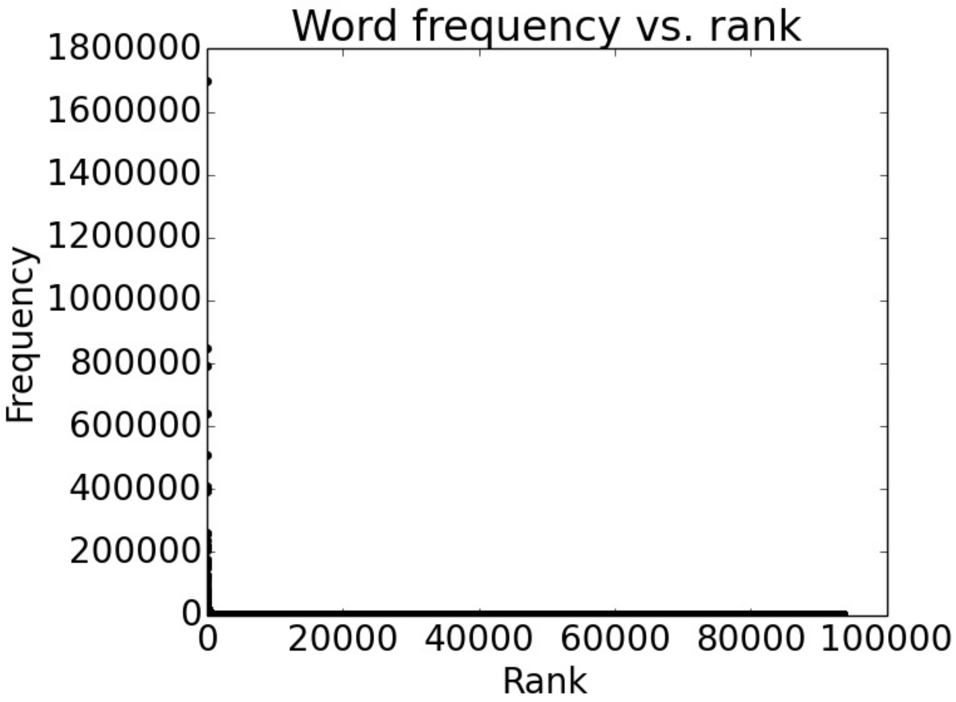
Word Counts

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

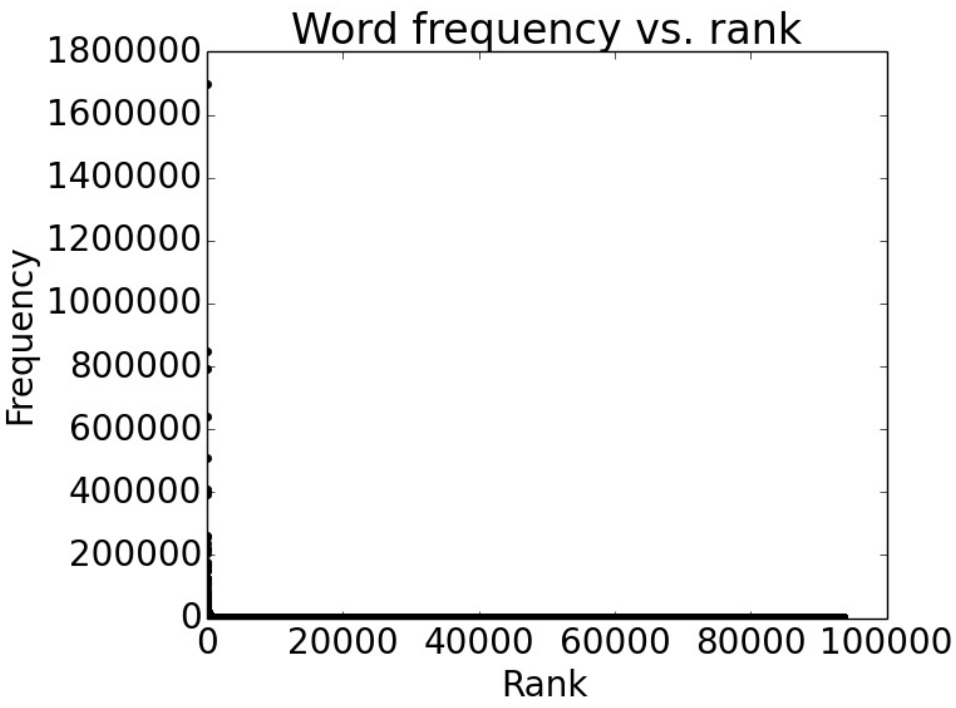
Plotting word frequencies

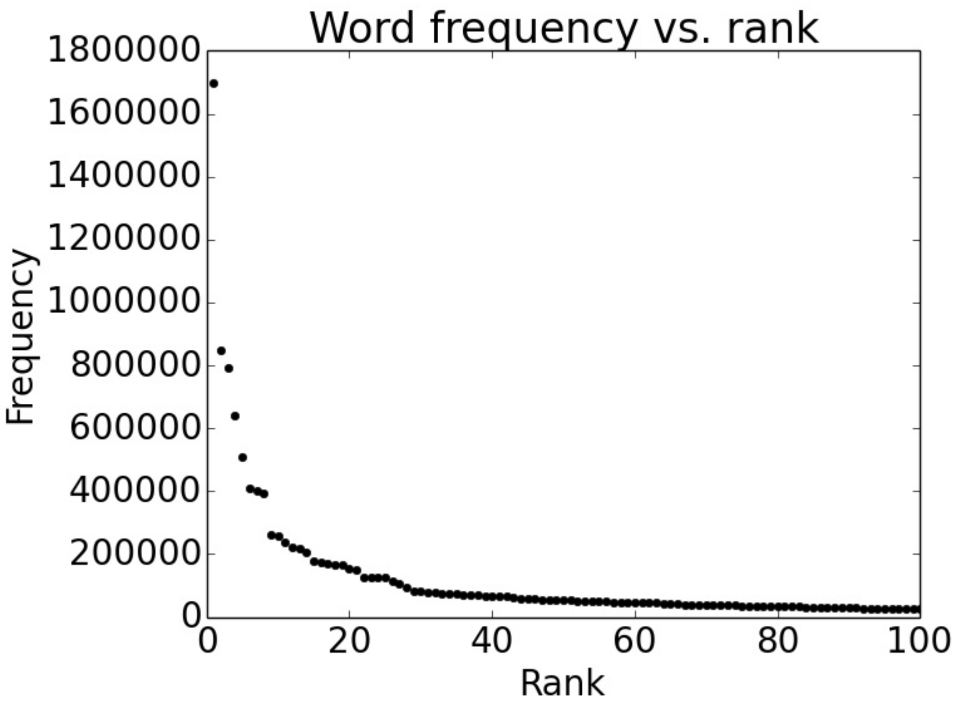
Order words by frequency. What is the frequency of nth ranked word?



Plotting word frequencies

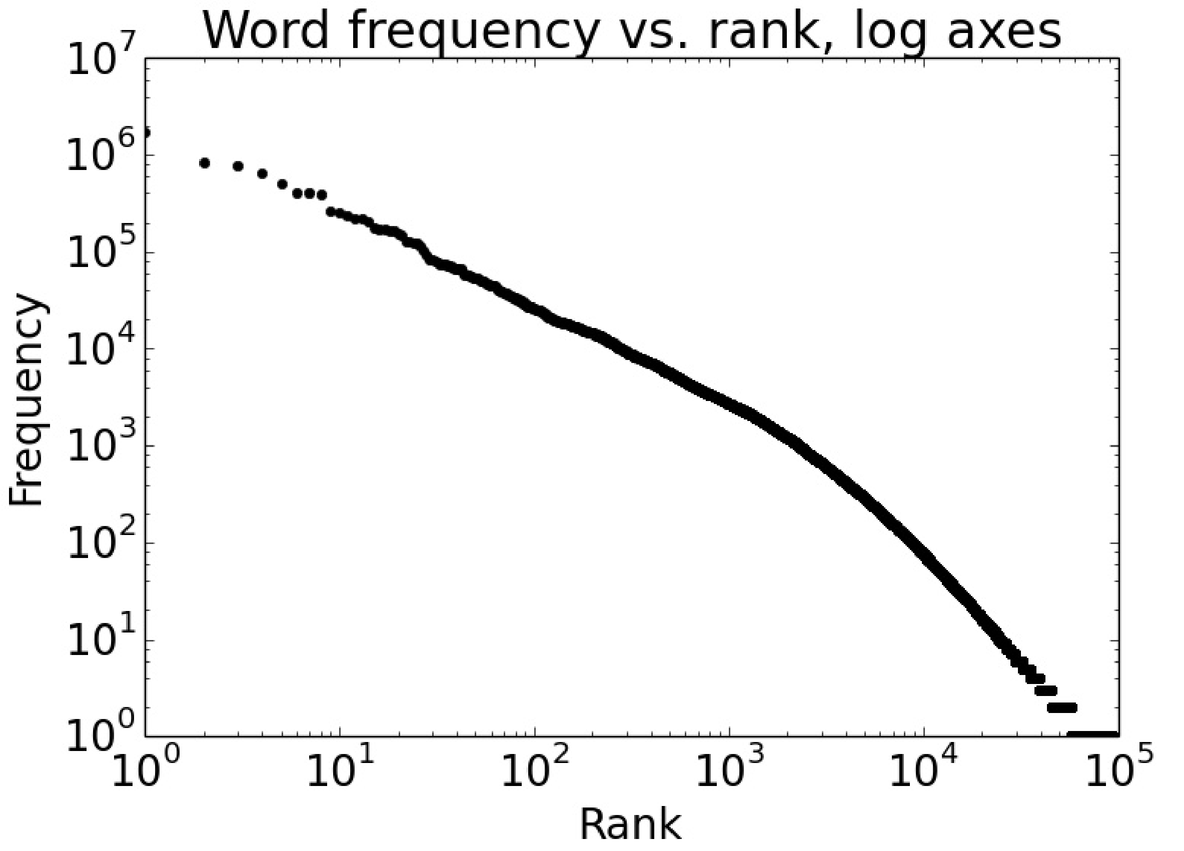
Order words by frequency. What is the frequency of nth ranked word?

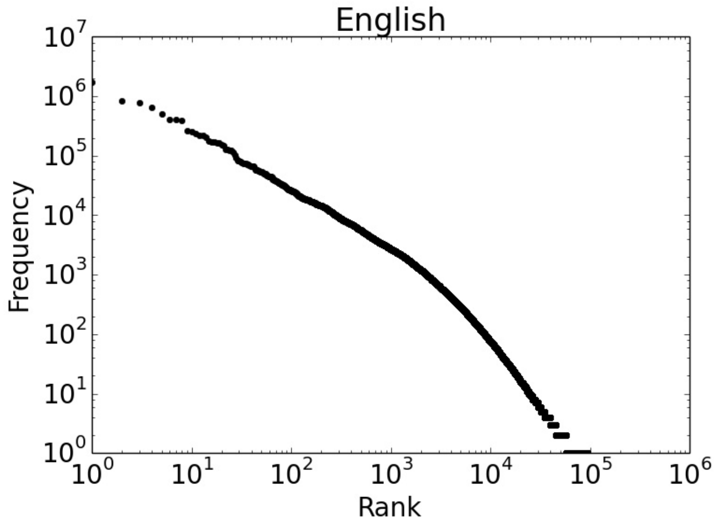


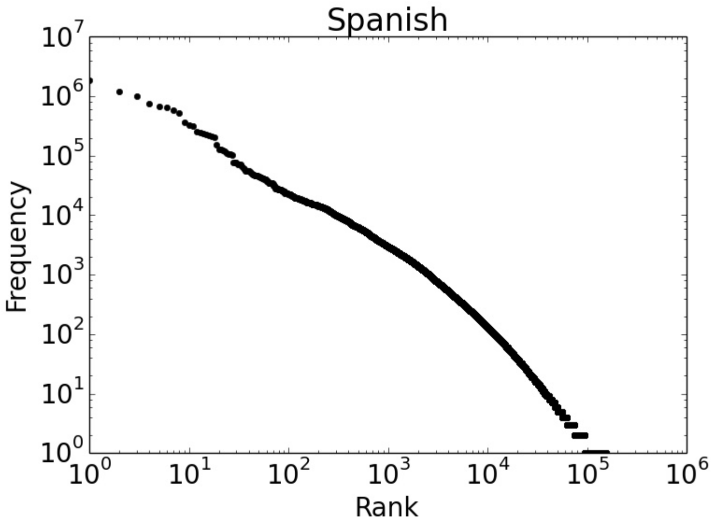


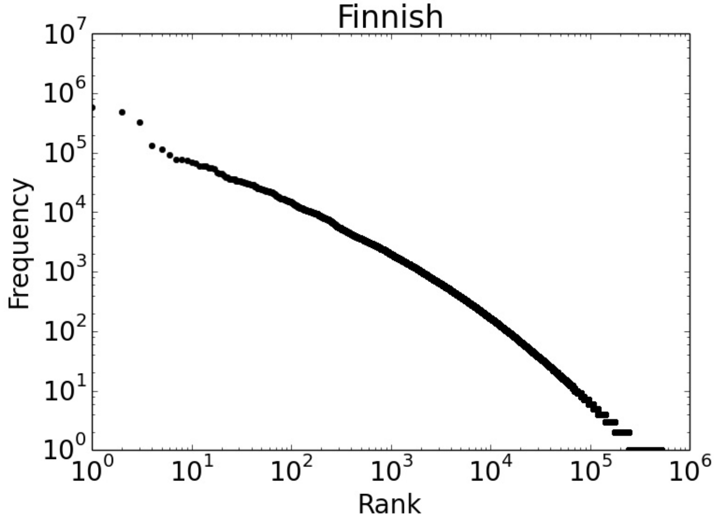
Rescaling the axes

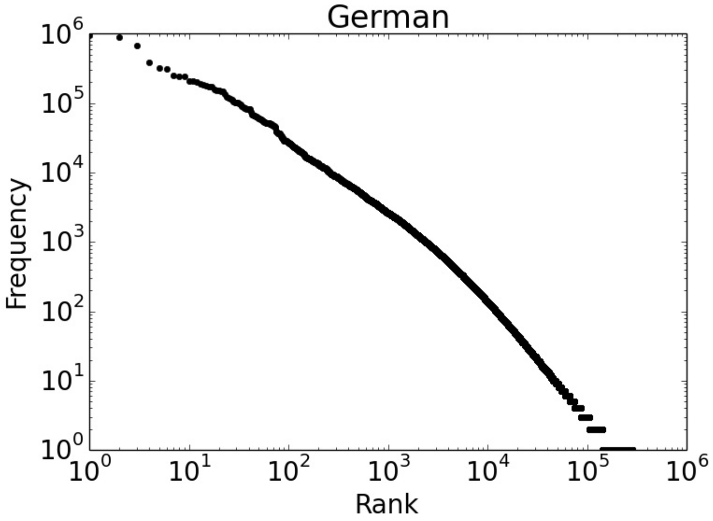
To really see what's going on, use logarithmic axes:




English
Frequency
Rank
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^0
10^1
10^2
10^3
10^4
10^5
10^6


Spanish
Frequency
Rank
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^0
10^1
10^2
10^3
10^4
10^5
10^6


Finnish
Frequency
Rank
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^0
10^1
10^2
10^3
10^4
10^5
10^6


German
Frequency
Rank
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^0
10^1
10^2
10^3
10^4
10^5
10^6

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?

Organization of Topics (pre-midterm)

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** (≈4 lectures): Getting everyone onto the same page with the fundamentals of text processing (Python 3/Unix) and linguistics.

- **N-grams** (≈2 lectures): Statistical modeling of words and word sequences.

- **Classification, Lexical Semantics with Classical Approaches** (≈2 lectures): Classifying documents or words without using grammatical structure. WordNet resource, classical ML methods.

- **Sequential Prediction with Classical Approaches** (≈5 lectures): Techniques that assign additional lingusitic information to words in sentences by modeling sequential relationships, including part-of-speech tagging and lexical semantic tagging.

Organization of Topics (post-midterm)

- **Language Modeling and Sequential Prediction with Vectors and Neural Networks** (≈3 lectures): Models for characterizing words and text collections based on unlabeled data, or nonlinear models (neural networks) without hand-engineered features; and overviews of language technologies for text such as machine translation and question answering.

- **Hierarchical Sentence Structure** (≈5 lectures): Tree-based models of sentences that capture grammatical phrases and relationships (syntactic structure), as well as structured representations of within-sentence semantic relationships.

- **Other Learning Paradigms and Applications** (≈4 lectures): Models for characterizing words and text collections based on unlabeled data, or nonlinear models (neural networks) without hand-engineered features; and overviews of language technologies for text such as machine translation and question answering.

Backgrounds

This course has enrollment from multiple programs!:

- Linguistics
- Computer Science
- possibly: Data Analytics; Biology

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.

What's *not* in this course

- Formal language theory
- Computational morphology
- Logic-based compositional semantics
- Speech/signal processing, phonetics, phonology

(But see next 2 slides!)

Some Related Courses as of Spring 2019 (1/2)

In Linguistics:

- Intro to NLP (Amir Zeldes, last semester)
- **Computational Linguistics with Advanced Python** (Liz Merkhofer, this semester)
- Signal Processing (Corey Miller, Fall 2017)
- Statistical Machine Translation (Achim Ruopp, Spring 2018)
- Dialogue Systems (Matt Marge, Fall 2018)
- Computational Corpus Linguistics (Zeldes, last semester)
- **Computational Discourse Models** (Zeldes, this semester)

Some Related Courses as of Spring 2019 (2/2)

In Computer Science:

- Statistical Machine Translation (Achim Ruopp, Spring 2018)
- Machine Learning (Mark Maloof, last semester)
- Automated Reasoning (Maloof, Fall 2017)
- **Deep Reinforcement Learning** (Grace Hui Yang, this semester)
- Dialogue Systems (Matt Marge, Fall 2018)
- Data Analytics (Lisa Singh, last semester)
- **Information Retrieval** (Nazli Goharian, this semester)
- Text Mining & Analysis (Goharian, Fall 2017)

Course organization

- Instructor: Nathan Schneider
- TAs: Austin Blodgett, Jakob Prange
- Lectures: MW 3:30–4:45, ICC 209A
- Web site: for syllabus, schedule (lecture slides/readings/assignments): `http://tiny.cc/enlp`
  - *Make sure to read the syllabus!*
  - No hard-copy textbook; readings will be posted online.
- We will also use Canvas for communication once enrollment is finalized.