Introduction

IN4080
Natural Language Processing

Yves Scherrer & Pierre Lison

Course overview

This course

An introduction to the most common tasks and methods in NLP

- Some statistical (non-neural) methods:
  - Naïve Bayes, logistic regression, sequence labeling, N-gram language models
- Introduction to neural NLP models:
  - Feedforward neural networks, word and document embeddings, and *transformers*
- Tasks:
  - Text classification, information extraction, speech processing, dialog systems and QA, machine translation
- Ethical considerations

Challenges (for us teachers)

- You have (very) different backgrounds
  - For some, there will be some repetition
  - Others may experience a steep learning curve
- Balance between non-neural vs neural methods
  - “Older” statistical techniques remain useful to get acquainted with basic principles (many NLP tasks are easier to understand with simpler models)
  - But we will of course also cover *transformers* and their use to train modern Large Language Models (LLMs)
  - There is also a dedicated course on neural methods for NLP next semester

Our expectations

- Programming:
  - Familiar with Python and Jupyter notebooks
  - Basic knowledge about machine learning
- Mathematics:
  - Probabilities, Logarithms
  - Linear algebra (vectors and matrices)
- Linguistics:
  - Basic knowledge of linguistic analysis

We may offer “refresher” sessions in mathematics and linguistics if needed.

Let’s do a quick poll…

https://www.menti.com/ Code: 3242 4735

- In what program are you currently enrolled?
- What is your previous knowledge of NLP and related fields?
- Would you be interested in refresher sessions?

Organization

- Lectures on Mondays (14:15 – 16:00)
  - **Teachers**: Yves or Pierre
  - Recordings and slides online
  - Some of Pierre's lectures will be replaced by video recordings
- Lab sessions on Tuesdays (12:15 – 14:00)
  - No recordings or slides
  - **Group teachers**: Helene + N.N.
  - Start **next week** (Aug 26): set up computing environment, refresher on Python data science stack
- Communication:
  - Discourse forum, e-mail, …

Evaluation

- 4 mandatory assignments
  - Practical programming assignments
  - Pass/fail grading, prerequisite for final exam

- Final written exam
  - December 4, 9:00 – 13:00
  - Closed book, mostly theoretical questions
  - Determines course grade

Preliminary schedule

Aug 18	Introduction [YS+PL]	
Aug 25	Language models, tokenization [YS]	**Oblig 1**: LMs and text classification *Deadline*: Sep 12
Sep 1	Text classification I [YS]	
Sep 8	Text classification II, vector semantics [YS]	
Sep 15	Sequence labelling [YS]	**Oblig 2**: sequence labelling and NNs *Deadline*: Sep 26
Sep 22	Neural networks [YS]	
Sep 29	Word embeddings, transformers [YS]	**Oblig 3:** LLMs and RAG *Deadline:* Oct 17
Oct 6	Large language models [YS/PL]	
Oct 13	Dialog systems I [PL]	
Oct 20	Dialog systems II, question answering [PL]	**Oblig 4:** dialogue systems and MT *Deadline:* Nov 7
Oct 28	Speech processing [PL]	
Nov 3	Machine translation [YS]	
Nov 10	Ethics I [PL]	
Nov 17	Ethics II, exam preparation [YS]	**Exam:** Dec 4

Syllabus

- Jurafsky & Martin: Speech and Language Processing, 3rd edition
  - https://web.stanford.edu/~jurafsky/slp3/
  - This book is a work in progress and thus only available online
  - We use the January 2025 draft

Programming setup

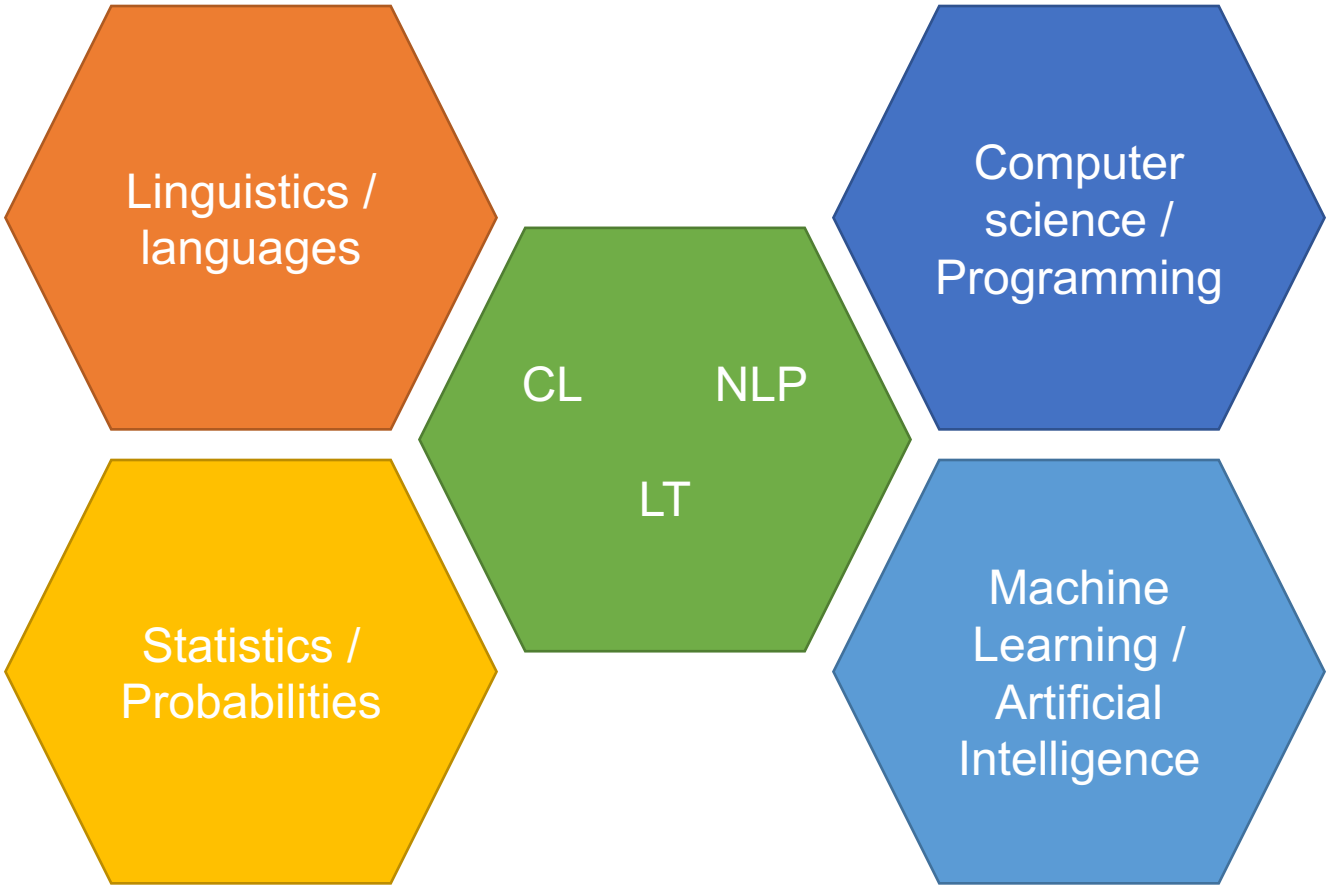
- Python + Jupyter Notebooks
- Data science stack:
  - Pandas, Numpy, Scipy, Matplotlib, …
- Machine learning libraries:
  - scikit-learn, PyTorch
- NLP-specific libraries:
  - spacy, nltk, transformers

What is NLP?

Names…

- Computational linguistics (CL)
  - Traditional name (cf. ACL)
  - Emphasizes applications that support (theoretical) linguistic research
- Natural language processing (NLP)
  - Originated in CS (natural vs. formal languages)
  - Emphasis on engineering, practical applications
  - Most often used name today
- Language technology (LT)
  - Sometimes used to combine CL and NLP (and speech processing)

An interdisciplinary field



Predominant approaches

Rule-based / knowledge-based

- 1950-2000
- Explicitly encode linguistic expert knowledge
- Difficult to scale, expensive

Statistic / data-driven / machine learning

- 1990-today
- Use machine learning methods to generalize from annotated data
- Division of labor between annotators / linguists and ML people
- Annotated datasets are costly to produce, but can easily be reused

Predominant approaches

Statistic / data-driven / machine learning

Non-neural models



Neural models

Task-specific data



Pretraining on task-independent data

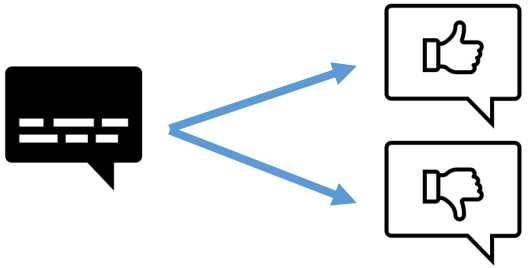
Main NLP tasks

Natural language generation



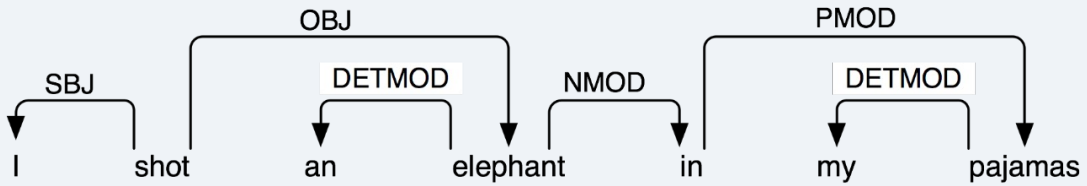
- Machine translation
- Question answering
- Grammatical error correction
- …

Annotation (natural language understanding)



- Hate speech detection
- Sentiment analysis
- Language identification
- Syntactic analysis
- …

The classical NLP pipeline

Split into sentences	Obama says he didn't fear for 'democracy' when running against McCain, Romney.
Tokenize (normalize)	| Obama | says | he | did| not | fear | for | ‘ | democracy | ‘ | when | running | against | McCain | , | Romney | .
Tag	Obama_N says_V he_PN did_V not_ADV fear_V …
Lemmatize	Says_V → say_V, did_V → do_V, running_V → run_V …
Parsing (dependency)	
Coreference resolution	Obama says he did not …..
Semantic relation detect.	Fear(Obama, Democracy) Run_against(Obama, McCain),..
Negation detection	… did not fear … → Not(Fear(Obama, Democracy))

This pipeline is less and less relevant nowadays:

- Most often, the goal is not to produce some kind of logical expression of the sentence meaning.
- Models can directly produce the lower levels without the upper ones.

Working with text data

Main focus of this course: written text in English

A dataset, as presented in a typical ML course

	spam	chars	lines breaks	'dollar' occurs. numbers	'winner' occurs?	format	number
1	no	21,705	551	0	no	html	small
2	no	7,011	183	0	no	html	big
3	yes	631	28	0			
4	no	2,454	61	0			
5	no	41,623	1088	9			
…							
50	no	15,829	242	0	no	html	small

Most data doesn't come as nicely formatted tables… How do we get from raw text to such tables?

Output class
Dependent variable

Input features
Explanatory/independent variables

A dataset, as presented in a typical ML course

Why exactly these features/columns?

	spam	chars	lines breaks	'dollar' occurs. numbers	'winner' occurs?	format	number
1	no	21,705	551	0	no	html	small
2	no	7,011	183	0	no	html	big
3	yes	631	28	0	no	text	none
4	no	2,454	61	0	no	text	small
5	no	41,623	1088	9	no	html	small
…							
50	no	15,829	242	0	no	html	small

How are these values defined?

Data preparation

- Preprocessing
  - Remove unwanted elements (empty lines, HTML tags…)
  - Split text into sentences
  - Tokenize (split sentences into “words” / tokens)
  - Case folding
- Task definition
  - Define scope of one instance: word, sentence, paragraph…
  - Define target labels

Sentence splitting

- How many sentences are in the following text?

  There are a number of ways this could happen, the churchmen pointed out, and here is an example. Last month in Ghana an American missionary discovered when he came to pay his hotel bill that the usual rate had been doubled. When he protested, the hotel owner said: "Why do you worry?"

- How would you detect sentence boundaries?
  - Split after .?!:
  - Maybe take into account capitalization
- Difficulties / corner cases?

Sentence splitting

Difficulties / corner cases?

- Is the colon (:) a sentence boundary?
- The full stop does not always mark sentence boundaries (e.g. abbreviations)
- But what about abbreviations at the end of a sentence?
- Titles typically do not end with a punctuation sign, but should be considered as sentences
- What about incomplete utterances (e.g. on social media)?

Sentence splitting

Sentence splitters are mostly rule-based (regular expressions):

- https://github.com/mediacloud/sentence-splitter
  - Look at the files in the non_breaking_prefixes folder
- SpaCy Sentencizer can use custom rules: https://spacy.io/api/sentencizer

Tokenization

- How many words/tokens does the following sentence contain?

  For example, this isn't a well-formed example.

  - Split at whitespace: 7
  - Separate punctuation (,.): 9
  - What should be done with hyphenated words?
    - well-formed or well | -formed or well | - | formed ?
  - What about contractions?
    - isn't or is | n't or isn | ' | t or is | not ?
- Tokenization is mostly a matter of conventions. There is generally not a single correct solution.

Tokenization

Examples of tokenization tools:

- As part of the Stanza toolkit: https://stanfordnlp.github.io/stanza/tokenize.html
  - Does sentence splitting and tokenization at the same time
- https://github.com/hplt-project/sacremoses
- https://spacy.io/api/tokenizer

Some languages don't use spaces to separate words. This makes tokenization much more complicated…

- E.g. Chinese – Jieba https://github.com/fxsjy/jieba

Case folding

- Should The be a different token than the?
- Should Apple be a different token than apple?

- Simplest solution:
  - Convert everything to lowercase
- Truecasing:
  - Collect a list of unique (lowercased) words
  - Count how often each word appears lowercased and capitalized
  - Replace all occurrences at the beginning of a sentence by the most frequent case

Working with text data



Look at the data

- Before and after preprocessing



A lot of things can go wrong here

- But they are usually easy to fix!



Ensure that:

- The character encoding is correct
- There is no unwanted markup
- The correct language is set for language-specific tools
- The same preprocessing is applied to all datasets

Statistical properties of text data

A quick glimpse at the “Computational Linguistics” side of the field...

Statistical properties of text data

- What can we do with a large corpus of text?
  - Count words, characters, word sequences, …
- With these counts, we can
  - Compare different corpora with each other
  - Discover statistical regularities (frequency laws)
  - Build statistical models that can generate new texts and classify existing texts

Count words

- The result is a **frequency distribution**
- Results depend on preprocessing
- What words do you expect to be most frequent?

Word Tally

the	𝍸 𝍸 𝍸 ||||
been	𝍸 𝍸 |
message	||||
persevere	|
nation	𝍸 |||

https://www.nltk.org/book/ch01.html

Count words

- These are the most common words in Tom Sawyer:

Chris Manning & Hinrich Schütze (1999): *Foundations of Statistical Natural Language Processing*, MIT Press.

Word	Freq.
the	3332
and	2972
a	1775
to	1725
of	1440
was	1161
it	1027
in	906
that	877
he	877
I	783
his	772
you	686
Tom	679
with	642

- What would the least common words be?
- Words that only occur once in a text are called **hapax legomena** (or hapaxes).

Lexical diversity

- **Tokens:** number of words in running text (with repetition)
- **Types:** number of unique words in running text
- "one cat caught five mice and three cats caught one mouse" → 11 tokens, 9 types
- **Type-token ratio (TTR):** types / tokens, expressed in %
  - 81.8% for the example sentence
  - TTR is a measure of **lexical diversity**
  - Which type of text has higher TTR, a newspaper article or a transcript of an informal conversation?

Collocations

- A **collocation** is a sequence of words that occur together unusually often.
- Examples of two-word collocations for two texts:

```
>>> text4.collocations()
United States; fellow citizens; four years; years ago; Federal
Government; General Government; American people; Vice President; Old
World; Almighty God; Fellow citizens; Chief Magistrate; Chief Justice;
God bless; every citizen; Indian tribes; public debt; one another;
foreign nations; political parties
>>> text8.collocations()
would like; medium build; social drinker; quiet nights; non smoker;
long term; age open; Would like; easy going; financially secure; fun
times; similar interests; Age open; weekends away; poss rship; well
presented; never married; single mum; permanent relationship; slim
build
```

https://www.nltk.org/book/ch01.html

Collocations

Some Norwegian examples (verb-preposition-noun):

	Verb	Prep	Noun	O/E-score	t-score	f(V)	f(P)	f(N)	f(VPN)
526391	løslate	mot	kausjon	4.26787e+11	12.2882	997	32429	198	151
288758	glimre	med	fravær	1.66425e+11	10.4403	154	211986	363	109
15345	annullere	for	offside	9.2411e+10	13.9642	468	131796	619	195
220461	fremstille	for	varetektsfengsling	8.24627e+10	17	901	131796	534	289
217539	framstille	for	varetektsfengsling	7.8356e+10	12.3693	502	131796	534	153
259066	føye	inn	rekke	7.70145e+10	15.3948	930	39183	1528	237
499020	lette	på	slør	5.53153e+10	13.2288	710	385759	209	175
88548	bortføre	fra	hjem	4.93984e+10	12	389	71064	1908	144
928314	virke	mot	hensikt	3.51002e+10	10.198	1999	32429	827	104
198343	forlate	uten	avtale	3.42333e+10	16.9706	3349	11403	3986	288
498296	lete	etter	olje	2.2257e+10	11.3578	1950	85229	631	129
13232	anke	til	lagmannsrett	2.0903e+10	10.8628	588	235379	738	118
566554	oppdatere	På	minutt	1.9798e+10	15.8114	946	22785	10600	250
604537	puste	i	nakk	1.91865e+10	10.5357	466	676598	332	111
568708	oppgi	i	mail	1.83119e+10	21.4476	985	676598	682	460

Zipf's law

The product of the frequency of a word f and its position in the frequency list (rank) r is constant:

$$f \cdot r = k$$

- The first word is twice as frequent as the second.
- The first word is three times as frequent as the third.
- …

George K. Zipf (1932): *Selected Studies on the Principle of Relative Frequency in Language*. Harvard, MA: Harvard University Press.

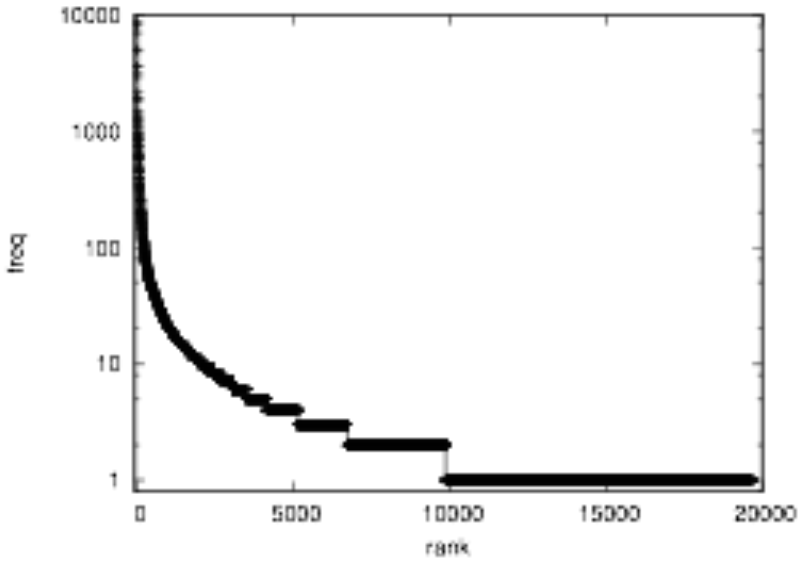
Zipf's law

Word	Freq. (f)	Rank (r)	$f \cdot r$	Word	Freq. (f)	Rank (r)	$f \cdot r$
the	3332	1	3332	turned	51	200	10200
and	2972	2	5944	you'll	30	300	9000
a	1775	3	5235	name	21	400	8400
he	877	10	8770	comes	16	500	8000
but	410	20	8400	group	13	600	7800
be	294	30	8820	lead	11	700	7700
there	222	40	8880	friends	10	800	8000
one	172	50	8600	begin	9	900	8100
about	158	60	9480	family	8	1000	8000
more	138	70	9660	brushed	4	2000	8000
never	124	80	9920	sins	2	3000	6000
Oh	116	90	10440	Could	2	4000	8000
two	104	100	10400	Applausive	1	8000	8000

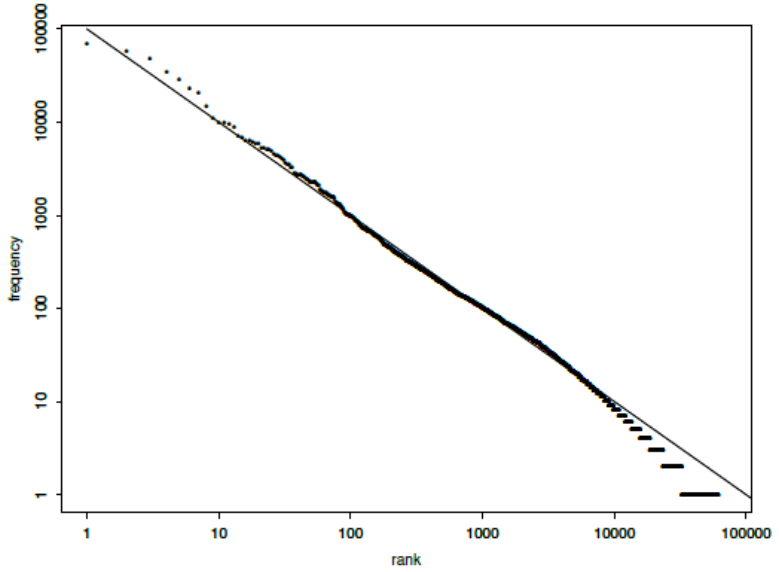
C. Manning & H. Schütze (1999): *Foundations of Statistical Natural Language Processing*, MIT Press, ch. 1.

Zipf's law

Visualization (log scale on Y-axis only):



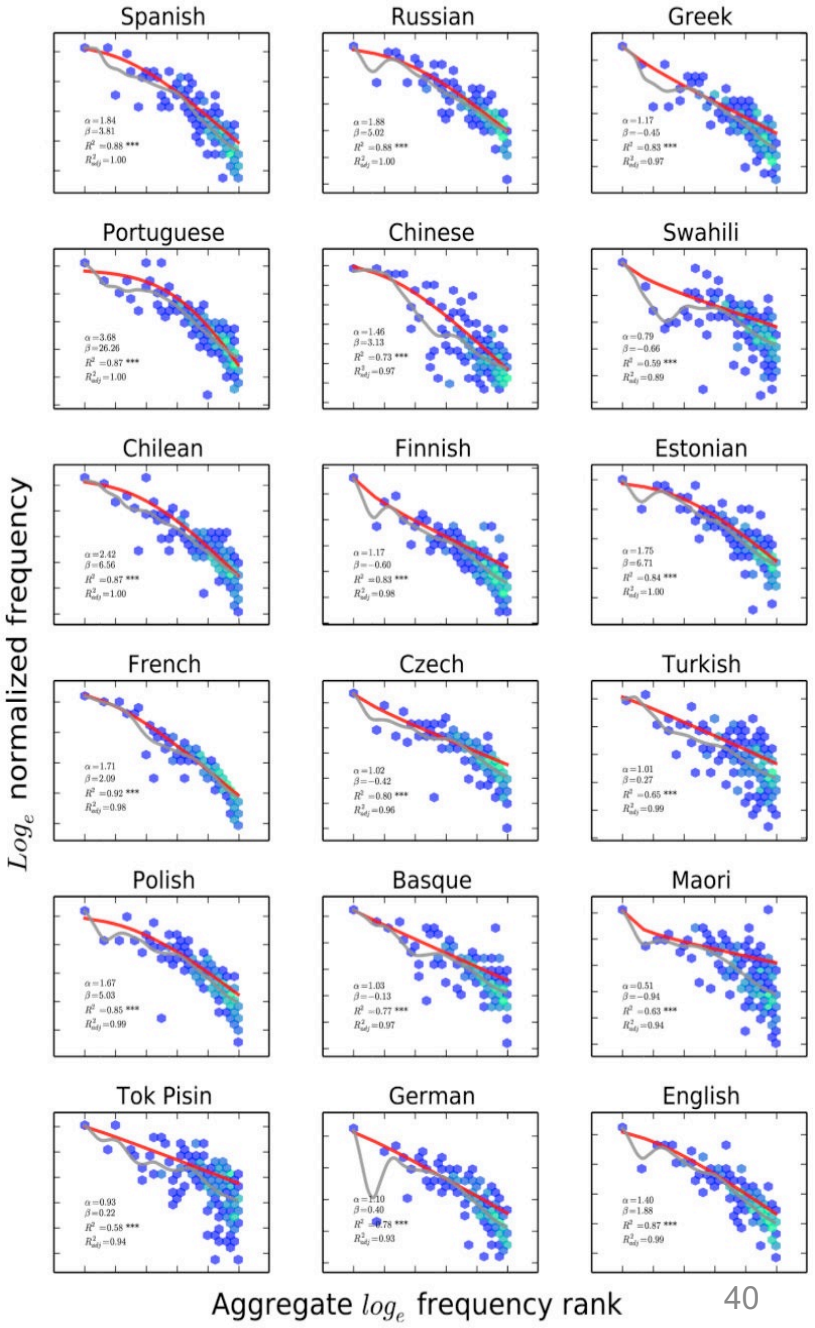
Visualization (log scale on X- and Y-axis):



Zipf's law

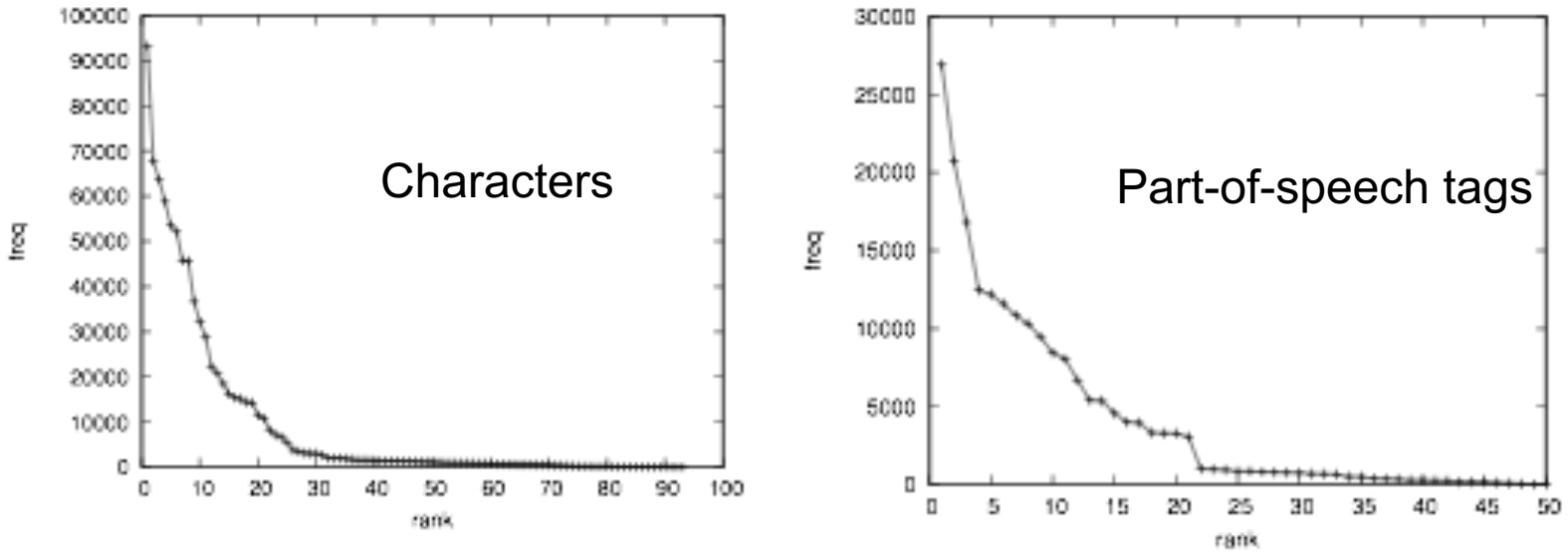
Is this only true for English?

Steven Piantadosi (2014): "Zipf's word frequency law in natural language: A critical review and future directions". *Psychon Bull Rev.* 21 (5): 1112–1130.



Zipf's law

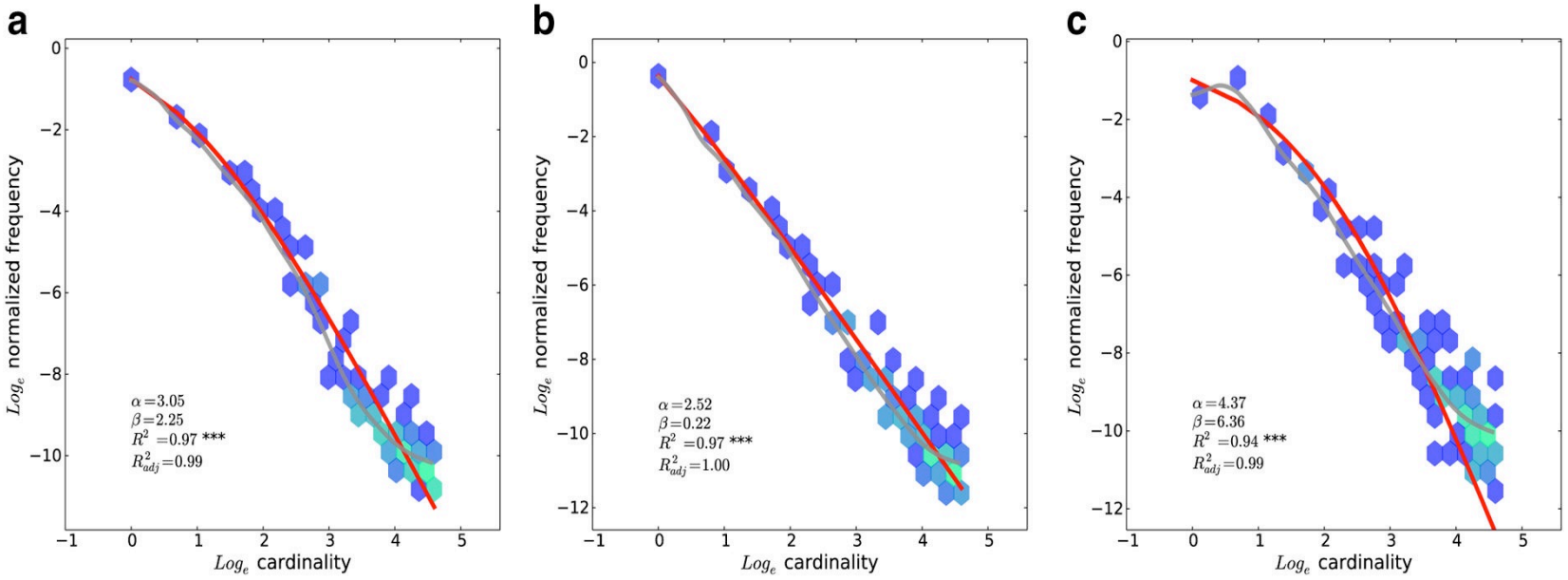
Is this only true for words?



(English)

Zipf's law

The same effect can be seen when looking only at numerals ("one", "two", ...) [Piantadosi 2014]:



English Russian Italian

Other laws by Zipf

- The number of **meanings** of a word is correlated with its frequency
- The **length** of a word is inversely correlated with its frequency
- Words tend to clump together in a text

Zipf's law – Conclusions

Zipf's law holds **approximately**:

- Unstable behavior for most frequent words
- Large variation for least frequent words
- The constant (slope of the regression line) differs according to language and data type

Zipf's law – Conclusions

"Psychological" interpretation:

- Zipf: Minimize effort of communication by using a small set of (easily accessible) common words and a large set of rare words (to prevent ambiguity)
- Piantadosi: *Word meaning is a substantial determinant of frequency, and it is perhaps intuitively the best causal force in shaping frequency. "Happy" is more frequent than "disillusioned" because the meaning of the former occurs more commonly in topics people like to discuss.*

Zipf's law – Conclusions

Practical implications:

- Almost all words are rare
- Whatever the size of the corpus, many words occur only once, and many words will not appear at all
  - If we train a model on a specific set of words, we need to make sure that they can handle words that are not included in this set

Readings

- Jurafsky & Martin, chapter 2:
  - §2.2 Words
  - §2.3 Corpora
  - §2.4, §2.5.1 Tokenization (skip §2.5.2 for now)
  - §2.7 Sentence segmentation

- Manning & Schütze, *Foundations of Statistical Natural Language Processing*
  - Chapter 1 on TTR, Collocations, Zipf's law etc.
  - PDF on the course page