# Introduction

## (SNLP tutorial)

Vilém Zouhar, Awantee Deshpande, Julius Steuer

27th, 29th April 2021

# Overview

- Hello
- Topics
- Requirements
- Materials
- Assignments
- Homework
- Zipf's Law
- QA

# Hello

Who am I?

# Hello

Who am I?

Who are you?

# Topics

- Language properties, Zipf's Law, Basic statistical formalism
- Information theory (Shannon's game, Code Length, Compression), Entropy
- Language modelling, Backing-off models (interpolation, discounting, smoothing)
- Text classification, Algorithms (kNN, Decision Trees, SVM, ...)
- Word Sense Disambiguation, Algorithms (Dictionary based-, translation-, Collocation-based)
- Information retrieval, Latent Semantic Analysis, Singular Value Decomposition
- Machine Translation, Word alignment
- POS Tagging, Named Entity Recognition
- ▸ Sequence labeling (Hidden Markov models, Conditional Random Fields)

# Requirements

## Tutorial Requirements (exam admission)

- 70% of mandatory points (~10 assignments, 10 points each)
- Tutorial points only for exam admission (no final grade influence)

## Tutorial Bonus Points

- ~2pts for extra exercises in the assignments
- 1pt for participating and *talking* in a tutorial
- Presenting a solution to the assignment (~5 points)
  - Let individual tutors known if you wish to present (in the respective tutor's channel)
  - Every group can present *at most* once

## Final Project

- 25% of the final grade
- Details TBD

## Transfer from last year

- Possible
- Do project and exam

# What's available

- Lectures by Prof. Klakow (recorded)
- Tutorials (not recorded, but allowed for private sharing)
- Corrected homework
- Consultations
  - Only in specific cases
  - By default **no** email and **no** personal chat
  - Ask questions during the lecture / tutorials
- Public forum (please use Piazza)
  - Ask questions
  - Other students will also benefit from the answers
  - You can answer someone else's issue

# Assignments

- Mandatory groups of 2
- Usually 3 exercises per assignment + a possible bonus question
- Jupyter notebook templates
  - Assignment + solution in the same notebook
  - Can use Google Colab or local runtime
  - Write solutions in Python files and import them
  - Submitted notebook must only contain your analysis and outputs
- Only one submission per group
  - Submit through Teams

# Dates / Times

- Lecture: Fridays 8:30-10:00
- Tutorials:
  - Awantee: Tuesday 14:15-15:45
  - Julius: Tuesday 12:15-13:45
  - Vilém: Thursday 16:00-17:30
- Assignments
  - Released (usually) Friday 23:59 (available in Teams)
  - Deadline (next) Friday 23:59 (submit in Teams)
- Exam: 23rd. Jul. (8:00-10:00)
- Project Deadline: Sometime in August

# Tutorial Content

- Review of the topics covered in class
- Presentation of the past assignment
- Discussing doubts in current assignment

# Current Homework

- Exercise 1: Notebook instructions
- Exercise 2: Stick breaking
- Exercise 3: Zipf's law at word level
- Bonus: Zipf's law at character level

# Languages

**Language**

$L \subseteq \Sigma^*$ (all possible substrings by elements of alphabet $\Sigma$)

# Languages

**Language**

$L \subseteq \Sigma^*$ (all possible substrings by elements of alphabet $\Sigma$)

- $\Sigma_1 = \{\texttt{a},\texttt{b},\ldots,\texttt{z},\texttt{ü},\texttt{ä},\texttt{ö}\}$
- $\Sigma_2 = \{\texttt{A},\texttt{G},\texttt{C},\texttt{T}\}$
- $\Sigma_3 = \{\texttt{def},\texttt{True},\texttt{:},\texttt{print},\ldots\}$
- $\Sigma_4 = \{\texttt{SELECT},\texttt{INSERT},\texttt{DROP},\ldots\}$
- $\Sigma_5 = \{\texttt{hallo},\texttt{ja},\texttt{nein},\ldots\}$
- $\Sigma_6 = \{\texttt{+},\texttt{-},\texttt{=},\texttt{1},\texttt{2},\texttt{3}\ldots\}$
- $\Sigma_7 = \{\texttt{+},\texttt{-},\texttt{=},\texttt{1},\texttt{2},\texttt{3}\ldots\}$

# Languages

**Language**

$L \subseteq \Sigma^*$ (all possible substrings by elements of alphabet $\Sigma$)

- $\Sigma_1 = \{\texttt{a},\texttt{b},\ldots,\texttt{z},\texttt{ü},\texttt{ä},\texttt{ö}\}$
- $\Sigma_2 = \{\texttt{A},\texttt{G},\texttt{C},\texttt{T}\}$
- $\Sigma_3 = \{\texttt{def},\texttt{True},\texttt{:},\texttt{print},\ldots\}$
- $\Sigma_4 = \{\texttt{SELECT},\texttt{INSERT},\texttt{DROP},\ldots\}$
- $\Sigma_5 = \{\texttt{hallo},\texttt{ja},\texttt{nein},\ldots\}$
- $\Sigma_6 = \{\texttt{+},\texttt{-},\texttt{=},\texttt{1},\texttt{2},\texttt{3}\ldots\}$
- $\Sigma_7 = \{\texttt{+},\texttt{-},\texttt{=},\texttt{1},\texttt{2},\texttt{3}\ldots\}$

- '`Oberfläche`' $\in L_1$ (German words)
- '`..GATTCCAATCAG`' $\in L_2$ (DNA)
- '`while True: f()`' $\in L_3$ (Python)
- '`SELECT * FROM tbl;`' $\in L_4$ (SQL)
- '`Wie geht's dir?`' $\in L_5$ (German)
- '`4=5`' $\in L_6$ (arithmetics)
- '`1=2+=3333=`' $\in L_7$ (???)

# Languages

**Language**

$L \subseteq \Sigma^*$ (all possible substrings by elements of alphabet $\Sigma$)

- $\Sigma_1 = \{\texttt{a},\texttt{b},\ldots,\texttt{z},\texttt{ü},\texttt{ä},\texttt{ö}\}$
- $\Sigma_2 = \{\texttt{A},\texttt{G},\texttt{C},\texttt{T}\}$
- $\Sigma_3 = \{\texttt{def},\texttt{True},\texttt{:},\texttt{print},\ldots\}$
- $\Sigma_4 = \{\texttt{SELECT},\texttt{INSERT},\texttt{DROP},\ldots\}$
- $\Sigma_5 = \{\texttt{hallo},\texttt{ja},\texttt{nein},\ldots\}$
- $\Sigma_6 = \{\texttt{+},\texttt{-},\texttt{=},\texttt{1},\texttt{2},\texttt{3}\ldots\}$
- $\Sigma_7 = \{\texttt{+},\texttt{-},\texttt{=},\texttt{1},\texttt{2},\texttt{3}\ldots\}$

- '`Oberfläche`' $\in L_1$ (German words)
- '`..GATTCCAATCAG`' $\in L_2$ (DNA)
- '`while True: f()`' $\in L_3$ (Python)
- '`SELECT * FROM tbl;`' $\in L_4$ (SQL)
- '`Wie geht's dir?`' $\in L_5$ (German)
- '`4=5`' $\in L_6$ (arithmetics)
- '`1=2+=3333=`' $\in L_7$ (???)

Usually defined by the alphabet and production rules (Automata and Grammar).

# Zipf's Law

1. Sort words/entries by frequency f(x)
2. r(x) = rank = position in the sorted list
   - Then $f(x) \propto \frac{1}{r(x)^\gamma}$ ($\gamma$ parameter)
3. Most common entry $m$.
   - Then $f(x) = \frac{f(m)}{r(x)^\gamma}$

# Zipf's Law

1. Sort words/entries by frequency f(x)
2. r(x) = rank = position in the sorted list
   - Then $f(x) \propto \frac{1}{r(x)^\gamma}$ ($\gamma$ parameter)
3. Most common entry $m$.
   - Then $f(x) = \frac{f(m)}{r(x)^\gamma}$

| Rank | Frequency | Predicted ($\gamma = 0.7$) |
|---|---|---|
| 4 | 606 | $1507/4^{0.7} = 571$ |
| 5 | 490 | $1507/5^{0.7} = 488$ |
| ... | | |
| 11 | 261 | $1507/11^{0.7} = 281$ |
| 12 | 252 | $1507/12^{0.7} = 264$ |

| Rank | Word [3] | Frequency |
|---|---|---|
| 1 | the | 1507 |
| 2 | and | 714 |
| 3 | to | 703 |
| 4 | a | 606 |
| 5 | of | 490 |
| 6 | she | 484 |
| 7 | said | 416 |
| 8 | it | 346 |
| 9 | in | 345 |
| 10 | was | 328 |
| 11 | I | 261 |
| 12 | you | 252 |
| 13 | as | 237 |
| 14 | Alice | 221 |

# Zipf's Law

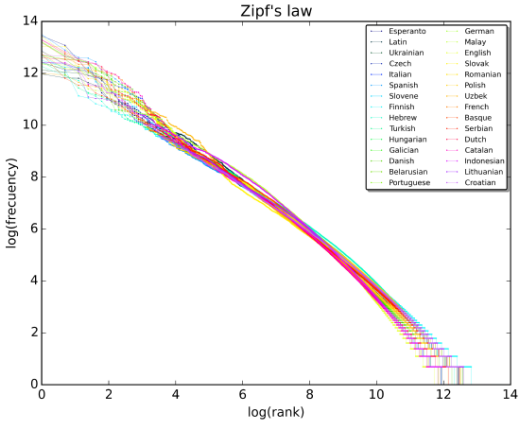


Figure 1: log-log plot of Zipf's Law applied to natural languages [4]

We are plotting:

$$\log\left(\frac{C}{\exp(x)^{\gamma}}\right)$$

$$\log(C) - \gamma \log \exp(x)$$

$$\log(C) - \gamma x$$

# Zipf's Law Notes

- Works beyond natural languages

| Rank | City | Population |
|---|---|---|
| 1 | New York, N.Y. | 8,491,079 |
| 2 | Los Angeles, Calif. | 3,928,864 |
| 3 | Chicago, Ill. | 2,722,389 |
| 4 | Houston, Tex. | 2,239,558 |
| 5 | Philadelphia, Pa. | 1,560,297 |
| 6 | Phoenix, Ariz. | 1,537,058 |
| 7 | San Antonio, Tex. | 1,436,697 |
| 8 | San Diego, Calif. | 1,381,069 |
| 9 | Dallas, Tex. | 1,381,069 |

# Zipf's Law Notes

- Works beyond natural languages
- DNA subsequences of fixed lengths

| Rank | City | Population |
|---|---|---|
| 1 | New York, N.Y. | 8,491,079 |
| 2 | Los Angeles, Calif. | 3,928,864 |
| 3 | Chicago, Ill. | 2,722,389 |
| 4 | Houston, Tex. | 2,239,558 |
| 5 | Philadelphia, Pa. | 1,560,297 |
| 6 | Phoenix, Ariz. | 1,537,058 |
| 7 | San Antonio, Tex. | 1,436,697 |
| 8 | San Diego, Calif. | 1,381,069 |
| 9 | Dallas, Tex. | 1,381,069 |

# Zipf's Law Notes

- Works beyond natural languages
- DNA subsequences of fixed lengths
- Code (programming languages)

| Rank | City | Population |
|---|---|---|
| 1 | New York, N.Y. | 8,491,079 |
| 2 | Los Angeles, Calif. | 3,928,864 |
| 3 | Chicago, Ill. | 2,722,389 |
| 4 | Houston, Tex. | 2,239,558 |
| 5 | Philadelphia, Pa. | 1,560,297 |
| 6 | Phoenix, Ariz. | 1,537,058 |
| 7 | San Antonio, Tex. | 1,436,697 |
| 8 | San Diego, Calif. | 1,381,069 |
| 9 | Dallas, Tex. | 1,381,069 |

# Zipf's Law Notes

- Works beyond natural languages
- DNA subsequences of fixed lengths
- Code (programming languages)
- Population of cities (frequency is city population)

| Rank | City | Population |
|---|---|---|
| 1 | New York, N.Y. | 8,491,079 |
| 2 | Los Angeles, Calif. | 3,928,864 |
| 3 | Chicago, Ill. | 2,722,389 |
| 4 | Houston, Tex. | 2,239,558 |
| 5 | Philadelphia, Pa. | 1,560,297 |
| 6 | Phoenix, Ariz. | 1,537,058 |
| 7 | San Antonio, Tex. | 1,436,697 |
| 8 | San Diego, Calif. | 1,381,069 |
| 9 | Dallas, Tex. | 1,381,069 |

# Mandelbrot Distribution

$$f(x) = \frac{m}{(c + r(x))^B}$$

Parameters to determine:

- m - Normalising term
- $c \geq 0$
- $B > 1$

# Resources

1. Course Website: https://www.lsv.uni-saarland.de/statistical-natural-language-processing-summer-2021/
2. Piazza: https://piazza.com/uni-saarland.de/spring2021/snlp
3. Teams channel: https://teams.microsoft.com/l/channel/19%3ac27c6662bea5450ebd05aad405690742%40thread.tacv2/General?groupId=4aab09db-6c09-4284-bfb0-2a94317c1d57&tenantId=67610027-1ac3-49b6-8641-ccd83ce1b01f