Recap

- Regular Expressions
  - Method to specify large sets of strings quickly
  - Combination of various special characters
  - Can be used in grep (and all programming languages)
- Concordances
  - Sometimes also called »keyword in context«
  - Table with a search query and left and right context
- Lecture
  - Probability theory
  - Pointwise Mutual Information (PMI) for collocation detection

Last Exercise

Let's extract a concordance (from poe or any other text)!

- Insert a space before each line end
- Remove all line breaks
- Unify all space to be a single space
- Feed the output into `grep -o` and inspect the concordance
- Our query includes the context in characters. Can you extend it such that we get tokens?

Query Ideas

- How does Poe write about men and women, how about cats and dogs?
- How did he use colors, e.g. red and green? What are things that are red, which things are green?
- Poe is a known horror author. Does he use the word »fear« as a noun or verb? In which contexts?

Processes, tmux, nano, and our first neural network

Sprachverarbeitung (VL + Ü)

Nils Reiter

April 25, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Organisatorisches

Nächste Woche

- Keine Übung (am Dienstag)
- Vorlesung (Donnerstag) findet regulär statt

Grund: Berufungsvorträge am 2. und 3. Mai

Slide from April 4

Why?

- Powerful: Many »small tasks« can be done directly on the command line
  - Without writing a full-fledged program for it
- Available: Every computer offers a command line as the most basic way of accessing it
- Economic: No overhead compared to GUIs
  - You can get the full machine performance
  - This also makes it networkable
- Simple: Developing GUIs is hard and takes a lot of time
  - Research software cannot afford this
  - User interface on the command line is easy to do
    - In fact: We have done this already in Java 1

Performance

- Has not really been an issue
- Two – related – aspects
  1. Time: How fast we get results
     - Depends on our budget, and how long we are willing to wait
  2. Space: How much memory we need in the process
     - Depends on our budget, and how we code things

Performance

- Has not really been an issue
- Two – related – aspects
  1. Time: How fast we get results
     - Depends on our budget, and how long we are willing to wait
  2. Space: How much memory we need in the process
     - Depends on our budget, and how we code things

Random Access Memory (RAM)

- Not: Disk space
- Strict upper bound (except for swap space, but that's very slow)
- `compute.spinfo`: 16 GB RAM

htop

- We need to find out how much memory our program consumes
- The tool `$ htop` can show us
  - Add the option `-u` to only show user processes
  - Press Q to quit
- Simpler alternative: `top`

demo

Time

- Many interesting processes take time
  - E.g., a week
- Exiting the connection terminates all processes
  - With pure SSH, we would need to keep the connection alive for the entire time – and if our ISP reconnects our DSL connection, we need to start again

Time

- Many interesting processes take time
  - E.g., a week
- Exiting the connection terminates all processes
  - With pure SSH, we would need to keep the connection alive for the entire time – and if our ISP reconnects our DSL connection, we need to start again
- Start our processes such that they continue even if we log off
  - Needs to be done before starting the process
  - Various options. Ours: `tmux`

Terminal Multiplexer – tmux

- A powerful tool »between ssh and the terminal«
- Start a new tmux session: `$ tmux`
- Attach to an existing session: `$ tmux at`
- Detach from a session: ctrl + b d
  - ctrl + b enters tmux control mode
  - If we detach, the session continues to run!
  - And we can re-attach to the session any time and from any where

https://tmuxcheatsheet.com

Text Editing

- We often need to edit plain text files via the command line
  - E.g. configuration or code
- Writing a regular expression and applying sed would work, but is cumbersome

Text Editing

- We often need to edit plain text files via the command line
  - E.g. configuration or code
- Writing a regular expression and applying sed would work, but is cumbersome
- There are multiple plain text editors we can use
  - vi, emacs, nano, ed, vim, ...
  - Most simple: nano

Text Editing

- We often need to edit plain text files via the command line
  - E.g. configuration or code
- Writing a regular expression and applying sed would work, but is cumbersome
- There are multiple plain text editors we can use
  - vi, emacs, nano, ed, vim, ...
  - Most simple: nano

Nano – Basic Commands

- Launch: `nano FILENAME`
- ctrl + x : Exit
- ctrl + o : Save (= write out)
- ctrl + r : Open (= read file)
- Editing: Normal keyboard layout, arrow keys

demo

Section 1

Exercise

Exercise

- Launch a tmux session
- Copy the file `/teaching/summer-2023/sprachverarbeitung/training.py` into your directory (`sprachverarbeitung`)
- Train the model, note down it's performance
- Increase the number of numbers to compare, and let it run again
  - For this, you'll need to edit the file `training.py`
- Play around with the other parameters