



University of
Zurich^{UZH}

Institute of Computational Linguistics

Programmiertechniken in der Computerlinguistik I

1. Vorlesung:

Betriebssysteme, Unix-Kommandos, Datei-Formate und
Kodierung, Wildcards




University of
Zurich^{UZH}

Institute of Computational Linguistics

PCL-1 – Übersicht über die Vorlesung

- 2 Vorlesungen zu UNIX-Kommandos, grep und regulären Ausdrücken
- danach Einführung in die Programmiersprache Python



**University of
Zurich** ^{UZH}

Institute of Computational Linguistics

Literatur

zu UNIX und grep


Einführung für Linux / UNIX-Anfänger

https://www.univention.de/fileadmin/uni_pdf_dokumente/debian_buch2_kapitel.pdf

zu Python

Steven Bird, Ewan Klein, Edward Loper (2009):
Natural Language Processing with Python --
Analyzing Text with the Natural Language Toolkit.

Page 17



**University of
Zurich** ^{UZH}

Institute of Computational Linguistics

Erstes Etappenziel

Gegeben ein Jahrbuch des SAC (z.B. von 1930)

- Wieviele Wörter enthält dieses Buch?
- Wieviele Sätze enthält dieses Buch?
- Wieviele Sätze in DE, FR, EN, IT, RM enthält dieses Buch?
- Welches sind die häufigsten Wörter in diesem Buch?

→ diese Fragen können wir am Ende der ersten Stunde beantworten. 😊

Page 18



Spätere Etappenziele

- Wieviele Wörter in DE, FR, EN, IT, RM enthält dieses Buch?
- Wieviele Adjektive, Nomen und Verben enthält dieses Buch?
- Was sind die häufigsten Paare Adjektiv + Nomen in diesem Buch?
- Welches ist der längste Satz / das längste Wort in diesem Buch?



Betriebssysteme

1. Microsoft Windows (auf vielen Rechnertypen)
2. Apple Mac OS (nur auf Apple Rechnern)
3. Unix / Linux

Betriebssysteme sind wie Automarken.

Man kommt mit jedem Auto von A nach B, mit manchen Autos ist es jedoch bequemer, sparsamer, zuverlässiger als mit anderen.



**University of
Zurich** ^{UZH}


Institute of Computational Linguistics

Kommandofenster

Bei Windows, Mac OS, Linux

- Erlaubt das Starten eigener Programme.
- Erlaubt den Dialog mit dem Betriebssystem.

Page 21



**University of
Zurich** ^{UZH}

Institute of Computational Linguistics

1. Schritt

Wo sind wir?

pwd

Wie kommen wir in das Verzeichnis, in dem wir arbeiten wollen?

cd Verzeichnis_X ODER **cd ..**

Wie sehen wir, welche Dateien im Verzeichnis vorliegen?

ls -l

Wie können wir in eine Datei hineinschauen?

more Datei_X.txt (quit with 'q'), [less]

Page 22



2. Schritt

Wie können wir die Wörter in einer Textdatei zählen?

wc Datei_X.txt

Wie können wir die Anzahl unterschiedlicher Wörter in einem Text zählen?

1. Text vertikalisieren
2. Wörter sortieren
3. Duplikate entfernen
4. Wörter zählen

Wie können wir einen Text vertikalisieren?

1. Suche Leerschlag und ersetze mit \n im Editor oder
2. **tr ' ' '\n' < Datei_X.txt**

Page 23



3. Schritt

Wie können wir die Zeilen einer Datei sortieren?

sort Datei_X.txt

Wie können wir eine Datei sortieren und identische Zeilen entfernen?

sort -u Datei_X.txt ODER
sort Datei_X.txt | uniq

Page 24



4. Schritt

Wie berechnen wir die Häufigkeit für jedes Wort in einer vertikalisierten Textdatei ?

sort Datei_X.txt | uniq -c

Wie sortieren wir nach der Häufigkeit?

sort -n (numerische vs. alphabet. Sortierung)

Wie sortieren wir rückwärts (= häufigste zuerst)?

sort -rn



5. Schritt

Wie können wir grosse Textmengen durchsuchen?

grep 'Suchwort' Dateimuster

Wieviele Sätze kommen im Jahrbuch 1900 vor?

grep '<s ' SAC_Jahrbuch_1900.xml | wc

Wieviele Sätze kommen in allen Jahrbüchern des 19. Jhd. vor?

grep '<s ' SAC_Jahrbuch_18*.xml | wc

grep -c '<s ' SAC_Jahrbuch_18*.xml



6. Schritt

Wie oft kommt das **Wort** 'Berg' vor?

```
grep 'Berg\t' *18*.xml | wc
```

Wie oft kommt das **Lemma** 'Berg' vor?

```
grep '\tBerg$' *18*.xml | wc
```

Wieviele Substantive kommen vor?

```
grep '\tNN\t' *18*.xml | wc
```

Wieviele deutsche Adjektive kommen vor?

```
grep '\tADJA\t' *18*.xml | wc
```



tr = transliterate

Vertikalisieren eines Textes: Ersetze jeden Leerschlag mit einem Zeilenumbruch

```
tr ' ' '\n' < Test1.txt
```

Konvertiere alle Grossbuchstaben in Kleinbuchstaben

```
tr '[A-Z]' '[a-z]' < Test1.txt
```



Unix-Kommandos

Umlenken der Ausgabe mit Überschreiben: '>'

Umlenken der Ausgabe mit Anfügen: '>>'

Umlenken der Eingabe: '<'

Verbinden von zwei Kommandos: Pipe '|'



Wildcards

= Stellvertretersymbole

- Wildcard * == beliebige Anzahl Zeichen
- Wildcard ? == ein beliebiges Zeichen

Wildcard [] == definiert eine Menge von Zeichen, von denen eines zutreffen muss.

Wildcard - == ein Strich in einer Klammer [] definiert einen Bereich von – bis.



Editor

Editor = Ein Programm zum Erstellen von Dateien.

Text-Editor = Ein Programm zum Erstellen von Text-Dateien.

Empfohlene Editoren:

- Mac: TextWrangler
- Windows: Notepad++
- Linux: Emacs ??



Dateiformate (für Textdateien)

Geschlossene Dateiformate: (schlecht für CL ☹)

- .doc (und auch .docx !?)
- .xls
- .pdf

Offene Dateiformate: (gut für Computerlinguistik ☺)

- .txt
- .xml (und .html)



University of
Zurich^{UZH}

Institute of Computational Linguistics

Zeichen-Kodierung

Latin-1 = iso-8859-1 (Umlaute als 1 byte)

Alt! ☹

UTF-8 (Umlaute als 2 bytes)

Gut! ☺

Unix-Kommando zur Bestimmung der Kodierung

file -i filename

Unix-Kommando zur Konvertierung einer Kodierung in
eine andere (**f**rom – **t**o)

iconv -f -t filename

Page 33