

Die Repräsentation und Auflösung von ambigen Wortbedeutungen in der Computerlinguistik

UNIX

PD Dr. Sabine Schulte im Walde

Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart

16. Januar 2010

- Betriebssystem
- UNIX stellt mächtige Werkzeuge für die Analyse und Bearbeitung natürlicher Sprache zur Verfügung

- Kommandos werden in der Zeile nach dem Prompt eingegeben.
 - erstes Wort = Kommando-Name = Programm-Name
 - zweites bis letztes Wort = Parameter, die dem Programm übergeben werden
- Arten von Parametern:
 - Optionen, die den Verlauf des Programms beeinflussen
 - Argumente, die vom Programm benötigt werden

- Eine Datei unter UNIX ist eine Sequenz von Zeichen.
- Eine Datei hat einen Namen und einen Typ.
- Dateien können Programme, Datensätze oder Texte enthalten.
- Um Dateien anzusprechen, benötigt man einen Pfadnamen (↪ Verzeichnis-Ort) und einen Dateinamen (↪ Datei).

- **ls** (list): zeigt Verzeichnis-Inhalt an

```
ls [-l(ong)a(11)] <datei>
```

ls

ls verzeichnis

- **cd** (change directory): Wechsel in anderes Verzeichnis

```
cd <verzeichnis>
```

cd beispiel-verzeichnis

- **cp** (copy): Dateien oder Verzeichnisse kopieren

```
cp <quelle> <ziel>
```

cp beispiel.txt beispiel_neu.txt

- **cat**: Dateien aneinanderhängen und zur Standardausgabe schreiben

```
cat <datei>
```

```
cat beispiel.txt
```

```
cat beispiel1.txt beispiel2.txt
```

- **less**: seitenweise durch Datei blättern

```
less <datei>
```

```
less beispiel.txt
```

- **head/tail**: Anfang/Ende einer Datei ausgeben

```
head <datei>
```

```
tail <datei>
```

```
head beispiel.txt (standardmäßig: 10 Zeilen)
```

```
tail beispiel.txt
```

```
head -50 beispiel.txt (Änderung des Standards)
```

```
tail -50 beispiel.txt
```

- **wc** (word count): zählt die Zeilen, Wörter und Zeichen eines Textes

```
wc <datei>
```

```
wc beispiel.txt
```

Pipe

- Pipe (engl.) = Röhre
- Symbol: | (senkrechter Strich in der Befehlseingabe)
- Hilfsmittel, um mehrere Filter miteinander zu verbinden
- Der erste Filter erzeugt ein Ergebnis, das vom zweiten Filter weiterverarbeitet werden kann, der zweite Filter erzeugt ein Ergebnis, das vom dritten Filter weiterverarbeitet werden kann, usw.
- Beispiele:

```
ls verzeichnis-name | head -20
```

```
cat datei | less
```

- **less**: seitenweise durch Datei blättern

```
less <datei>
```

```
less beispiel.txt
```

- **grep**: durchsucht Dateien nach Ausdrücken

```
grep [-i(gnore)v(s)] <ausdruck> <datei>
```

```
grep 'Mädchen' beispiel.txt
```

```
grep 'chen' beispiel.txt
```

```
grep 'ung$' beispiel.txt
```

```
grep '^$' beispiel.txt
```

```
grep -v '^$' beispiel.txt
```

- **sort**: Zeilen von Dateien sortieren

```
sort [-n(umeric)r(everse)] <datei>
```

```
sort beispiel.txt
```

```
sort -nr beispielzahlen.txt
```

- **uniq**: aufeinanderfolgende doppelte Zeilen entfernen

```
uniq [-c(ount)] <datei>
```

```
sort beispiel.txt | uniq
```

```
sort beispiel.txt | uniq -c
```

Datei-Inhalte manipulieren

- **tr** (translate): Zeichen transliterieren oder löschen

```
tr [-c(omplement)d(elete)s(squeeze-repeats)]  
⟨ausdruck1⟩ ⟨ausdruck2⟩
```

```
cat beispiel.txt | tr 'a-z' 'A-Z'
```

```
cat beispiel.txt | tr -c '0-9a-zA-Z' 'n'
```

```
cat beispiel.txt | tr -s 'a-zA-Z'
```

- **sed** (stream editor): Editor zur nicht-interaktiven Textbearbeitung

```
sed ⟨ausdruck⟩ ⟨datei⟩
```

```
cat beispiel.txt | sed 5q
```

```
sed s/Apfel/Birne/g beispiel.txt
```

```
sed s/ung$/ /g beispiel.txt
```

- **man**: greift auf das Manual eines Befehls zu

```
man <befehl>
```

```
man cat
```

```
man head
```

- **info**: dito

```
info <befehl>
```

```
info cat
```

```
info head
```



Jens Vieler.

Einführung in UNIX.

Technical Report A/037/9202, Universitätsrechenzentrum,
FernUniversität Gesamthochschule in Hagen, 1992.



Kenneth W. Church.

Unix for Poets.

Manuscript, AT&T Research, 1994.



Chris Brew and Marc Moens.

Data-intensive linguistics.

Manuscript, 2002.