

Abfragen, Emu, und Emu-R Beziehungen

(Datenbank votexample)

Emu starten, Database Operations -> Query Database

Add to list *

um Abfragen aus allen Äußerungen durchzuführen

Alle [d] Verschlüsse

phonetic = d

Als Segmentliste speichern (z.B. d.txt) in einem Verzeichnis (z.B. pfad)

R starten

```
library(emu)
```

```
# Segmentliste einlesen
```

```
read.emusegs(file.path(pfad, "d.txt"))
```

```
# Oder
```

```
emu.query("votexample", "*", "phonetic = d")
```

Einfache Abfragen

```
word = Dachlatte
```

```
phonetic = d | t
```

```
phonetic != xxx
```

Reihenfolgen-Abfragen (NB: immer aus derselben Ebene)

```
[phonetic = d -> phonetic = H]
```

```
[#phonetic = d -> phonetic = H]
```

```
[phonetic = d -> #phonetic = H]
```

```
[phonetic = d | t -> phonetic = H]
```

Hierarchische Abfragen

```
[phonetic = d ^ word = Dach]
```

```
[phonetic = d ^ word = Dachs | Dachlatte]
```

```
[phonetic = x ^ #utterance !=xxx]
```

Parallel-Abfragen

```
cano !=xxx & word = Dachrinne
```

Reihenfolgen und hierarchische Abfragen mit emu.requery()

```
# Segmentliste von h
h.s = emu.query("votexample", "*", "phonetic = H")

# Die davorkommenden Segmente
ver.s = emu.requery(h.s, "phonetic", "phonetic", seq=-1)

# Wie oben aber nur die Etikettierungen davon
ver.l = emu.requery(h.s, "phonetic", "phonetic", seq=-1, j=T)

# Die Wörter davon
wort.s = emu.requery(h.s, "phonetic", "word")

# Die kanonischen Transkriptionen
cano.s = emu.requery(wort.s, "word", "cano", j=T)
```