

Aufgabe B : F0-Gipfel-Position in Reim /Vn/ mit gespannten und ungespannten Vokalen in initialer(I) oder finaler (F) Phrasenposition

Template-Dateien florian.tpl, raphael.tpl, sabine.tpl in S:/IPSK/dbtemplates nach Ihrem Template-Verzeichnis kopieren

Datenbank: raphael

Lösung 1

Segmentliste vom n

n.s = emu.query("raphael", "*", "Phonetik=n")

Noff = end(n.s)

Eventliste vom Target

targ.s = emu.requery(n.s, "Phonetik", "f0Peak")

G = start(targ.s)

Segmentliste vom davorkommenden Segment (Vokal)

e.s = emu.requery(n.s, "Phonetik", "Phonetik", seq=-1)

Von = start(e.s)

$$p = (G - \text{Von}) / (\text{Noff} - \text{Von})$$

Labels der Phrase

p.l = emu.requery(n.s, "Phonetik", "Phrase", just=T)

table(p.l)

Zwei Labels korrigieren

temp = p.l != "I" & p.l != "F"

sum(temp)

p.l[temp] = "I"

Gesp-Labels vom Vokal

gesp.l = emu.requery(e.s, "Phonetik", "Gesp", just=T)

table(gesp.l)

labs = paste(gesp.l, p.l, sep = ".")

table(labs)

boxplot(p ~ labs)

Lösung 2

Phrasen kompl. einlesen und Von und Noff ermitteln

alles.times=emu.query("raphael", "*", "Phrase!=x")

Von=start(alles.times)

Noff=end(alles.times)

#Startzeitpunkte ermitteln

G=start(emu.requery(alles.times, "Phrase", "f0Peak"))

Aber bei Wert 17 ist ein Problem

```
# Formel berechnen
```

```
p = (G - Von)/(Noff - Von)
```

```
# Wert 17 ist unendlich
```

```
# Gesp. Werte abfragen
```

```
alles.gesp=emu.query("raphael", "*", "Gesp=j|n")
```

```
# Besser (sicherer)
```

```
alles.gesp = emu.requery(alles.times, "Phrase", "Gesp")
```

```
gesp.lab = label(alles.gesp)
```

```
gesp.lab = substring(gesp.lab, 1, 1)
```

```
# Die jeweiligen p Werte ermitteln
```

```
Gesp.Fin=p[label(alles.times)=="F"|label(alles.gesp)=="j"]
```

```
Gesp.In=p[label(alles.times)=="I"|label(alles.gesp)=="j"]
```

```
Ung.Fin=p[label(alles.times)=="F"|label(alles.gesp)=="u"]
```

```
Ung.In=p[label(alles.times)=="I"|label(alles.gesp)=="u"]
```

```
# Besser mit parallelen Vektoren vorzugehen
```

```
labs = paste(label(alles.times), gesp.lab, sep=".")
```

```
boxplot(p ~ labs)
```

```
# plotten
```

```
boxplot(Gesp.Fin,Gesp.In,Ung.Fin,Ung.In,ylab="Proportionale Dauer")
```

```
axis(1, 1:4, c("Gesp.Fin", "Gesp.In", "Ung.Fin", "Ung.In"))
```

```
Database: florian, 40 Aufnahmen
```

```
# alle Vokale
```

```
segvow=emu.query("florian", "*", "phonetic=i:|e:|a:|u:|y:|i|e|a|u|y")
```

```
# Oder
```

```
segvow = emu.query("florian", "*", "phonetic!=x & Start(word, phonetic)=1")
```

```
# onset der Vokale
```

```
start(segvow)
```

```
# alle Nasale
```

```
segn=emu.query("florian", "*", "phonetic=n")
```

```
# offset nasale
```

```
end(segn)
```

```
# alle Gipfel
```

```
segg=emu.query("florian", "*", "prosodic=G")
```

```
# Oder
```

```
segg2 = emu.requery(segvow, "phonetic", "prosodic")
```

```
start(segg)
```

```
# proportionale Position
```

```
prop_p=(start(segg)-start(segvow))/(end(segn)-start(segvow))
```

```
# vocal label
```

```

vow.l=label(seg vow)
# vector mit labeln 'gespannt' und 'ungespannt'
g_vow=c("i:", "e:", "a:", "u:", "y:")
u_vow=c("i", "e", "a", "u", "y")
vow.gu = vow.l
vow.gu[vow.l %in% g_vow]="gespannt"
vow.gu[vow.l %in% u_vow]="ungespannt"
# boxplot für gespannt und ungespannt
boxplot(prop_p ~ vow.gu)
# vector mit initial (I) und finaler (F) position in Phrase
segpos.l=label(emu.query("florian", "*", "word!=xxx"))
# Oder
segpos2.l= emu.requery(seg vow, "phonetic", "word", j=T)

```

```

# boxplot mit gespannt/ungespannt und I/F
boxplot(prop_p ~ segpos.l * vow.gu, ylab="Proportionale Dauer")

```