Siehe auch Harrington (2010), Kap. 4)

Die Beispiele dieser Abfragen in 1. und 2 beziehen sich auf die Sprachdatenbank aetobi

```
A. Grundlegende Abfragen
```

```
1. Einfache Abfragen
```

E = a

('a' der Ebene E)

'poisonous' der Wort Ebene

Word = poisonous

'bananas' und 'Anna' der Word-Ebene

Word = bananas | poisonous

Alle Tonakzente außer 'H*'

Tone != H*

Alle Tonakzente

Tone != xxx

Merkmale verwenden (siehe Legal Labels der aetobi Template-Datei)

Alle 'H*' und 'L*'

Tone = mono

2. Reihenfolge-Abfragen ->

'a b' der Ebene E

$$[E = a -> E = b]$$

'a' der Ebene E, und 'a' tritt auf vor 'b'

[#E = a -> E = b]

Alle Reihenfolgen von L* H* (L* gefolgt von H*)

[Tone = L^* -> Tone = H^*]

Alle Wörter, die nach 'the' vorkommen

[Word = the \rightarrow #Word !=x]

Die Beispiele dieser Abfragen in 3. und 4 beziehen sich auf die Sprachdatenbank ae

3. Parallel-Abfragen &

Alle Funktionswörter (also eine Segmentliste von 'the', 'in', 'am', 'it'...)

[Text!=x & Word = F]

4. Strukturelle Abfragen ^

(Ebenen E und F haben eine autosegmentelle oder hierarchische Beziehung)

 $E \rightarrow e \land F \rightarrow f$

e der Ebene e und e hat eine autosegmentelle oder hierarchische Beziehung zu f der Ebene F

```
Inhaltswörter assoziiert mit 'L-' der Ebene Intermediate [Word = C ^ Intermediate = L-]
```

Alle Wörter, dominiert von H- der Ebene Intermediate und L% der Ebene Intonational [[Text $!= x \land Intermediate = H-] \land Intonational = L\%$]

Alle Töne außer L* assoziiert mit 'L%' der Ebene Intonational [Tone $!= L^* \land Intonational = L\%$]

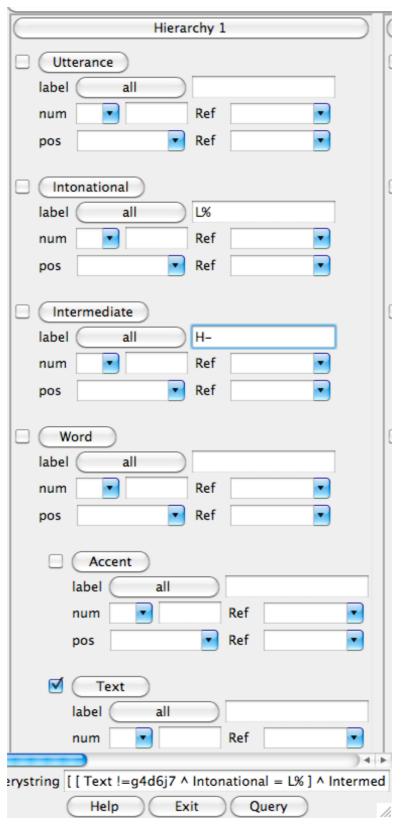
B. In Emu-R duplizieren

```
R starten, library(emu)
```

```
emu.query("aetobi", "*", "[Tone=mono]")
emu.query("aetobi", "*", "[ Word = bananas | poisonous]")
usw.
```

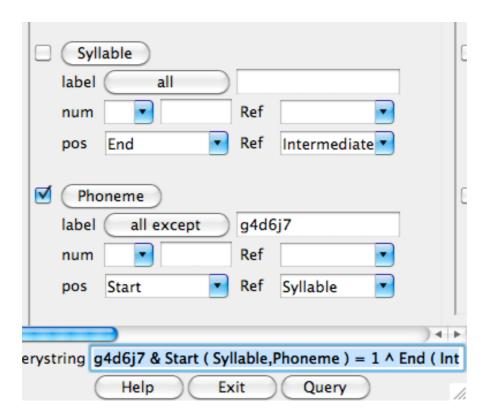
C. Graphical-User-Interface

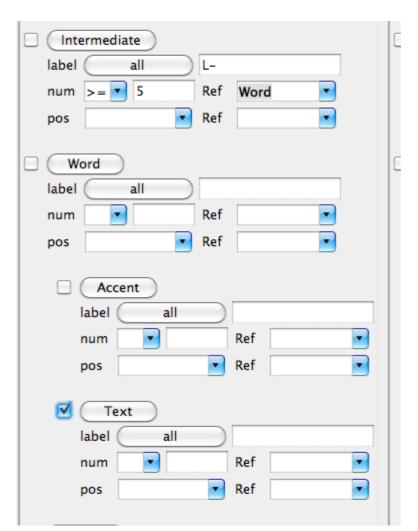
Alle Wörter, dominiert von H- der Ebene Intermediate und L% der Ebene Intonational (Abb. 1), nächste Seite.



D. Positions- und Anzahlabfragen

Silbeninitiale Phoneme in intermediärfinalen Silben.





L- Intermediärphrasen, die aus mindestens 5 Wörtern bestehen

E. Fragen

Erzeugen Sie ein Verzeichnis in Eigene Dateien, um Segmentlisten aus der **ae** Sprachdatenbank in den folgenden Fällen zu speichern.

- 1. 'his' Einheiten der Text Ebene
- 2. 'p', 't', und 'k' Einheiten der Phoneme-Ebene

3. Alle Wörter (der Text Ebene), die nach 'his' auftreten. 4. Eine Reihenfolge von "ei k" der Phoneme-Ebene 5. Alle Inhaltswörter (die Einheiten der Text-Ebene, die mit C der Word-Ebene assoziiert sind). 6. Die S-Einheiten der Accent-Ebene in der Reihenfolge W S 7. Eine Reihenfolge von zwei Einheiten der Text-Ebene: die erste Einheit ist 'the' (also Reihenfolgen wie 'the person' 'the situation' etc. 8. Alle 'ei' Einheiten der Phonem-Ebene in starken Silben (Syllable=S) 9. Schwache Silben assoziiert mit 'ei' Einheiten der Phonem-Ebene. 10. "m" und "n" Phoneme in wortinitialer Position 11. Intonationsphrasenfinale W-Silben 12. Wörter (Einheiten der Text-Ebene), die aus 3 Silben bestehen 13. Alle Phoneme in einsilbigen Wörtern 14. Wortfinale Phoneme in dreisilbigen Inhaltswörtern

Antworten auf der nächsten Seite

```
#1.
emu.query("ae", "*", "Text=his")
emu.query("ae", "*", "Phoneme=p | t | k")
emu.query("ae", "*", "[Text=his -> #Text!=x]")
emu.query("ae", "*", "[Phoneme=ei -> Phoneme = k]")
# 5.
emu.query("ae", "*", "[Text!=x & Word=C]")
# 6.
emu.query("ae", "*", "[Accent = W -> #Accent = S]")
# 7.
emu.query("ae", "*", "[Text = the -> Text !=x]")
# 8.
emu.query("ae", "*", "[Phoneme = ei ^ Syllable=S]")
emu.query("ae", "*", "[Phoneme = ei ^ #Syllable=W]")
emu.query("ae", "*", "[Syllable=W ^ Phoneme = ei]")
# 10.
emu.query("ae", "*", "[Phoneme= m | n & Start(Word, Phoneme)=1]")
emu.query("ae", "*", "[Syllable=W & End(Intonational, Syllable)=1]")
emu.query("ae", "*", "[Text!=x & Num(Text, Syllable)=3]")
# 13.
emu.query("ae", "*", "[Phoneme!=x ^ Num(Word, Syllable)=1]")
# 14.
emu.query("ae", "*", "[ [ Phoneme !=x & End ( Word,Phoneme ) = 1 ^{\land} Word = C ] ^{\land} Num (
Word, Syllable ) = 3 ]")
```