

Acht Dinge, die ein Praat-Skript (sehr) langsam machen:

0) Skripte mit geringen Datenmengen testen

Skripte erst mit einem File testen (und kontrollieren, ob Daten Sinn machen. Dann mit zwei Files in einer Directory/einem Folder testen, und erst dann auf größere Datenmengen anwenden.

1) Einen .wav-File öffnen, wenn man ihn gar nicht braucht:

Sollen nur Zeiten (Dauern oder Zeitpunkte) ausgewertet werden, braucht man nur die .TextGrid Files und nicht die .wav Files.

2a) Mehrfach das selbe berechnen

Kommandos/Funktionen wie “To Formant”, “To Pitch”, “To Intensity”, etc. berechnen die Daten immer für das gesamte Sound Objekt (i.d.R. ein .wav File). Wenn man mit einer (for/while/repeat) Schleife durch einen File geht und für jedes Segment, für das man die Daten braucht, jedesmal die Formanten/den Pitch/die Intensität berechnet, wird immer wieder das selbe für das gesamte Objekt berechnet.

Also: Vor der Schleife, die die Segmente durchläuft, einmalig die Formanten/den Pitch/die Intensität berechnen und dann die Daten für das spezielle Segment abfragen.

2b) Mehrfach das selbe berechnen

Wenn man Daten (Formanten etc.) nur für spezielle Segmente, z.B. ein [a] oder [i], braucht, ist es ineffizient, zuerst ein Skript für [a] laufen zu lassen, und dann ein zweites Mal für [i].

Also: bei der Ausgabe gleiche alle Segmente auflisten (bzw. mit ‘or’ in der ‘if’-Abfrage verknüpfen) für die man die Daten erheben will. Oder: die Daten für alle Segmente ausgeben lassen und später mit Excel im Ergebnisfile die entsprechenden Daten selektieren (bzw. sortieren und auswählen).

3) Nicht mehr gebrauchte Objekte nicht löschen (‘removen’)

Praat lädt alle Objekte (alles, was in der linken Liste steht) in den Hauptspeicher des Computers. Wenn das viele Daten sind (z.B. viele oder lange .wav Files, Spektren, Spektrogramme etc.) dann ist der Speicher irgendwann voll und das Betriebssystem fängt an, Daten auf die Disk auszulagern (‘swapen’), was alles etwa um den Faktor 1000 langsamer macht (und ev. das Skript crasht).

Also: Objekte (in den for/while/repeat Schleifen) löschen, wenn sie nicht mehr gebraucht werden (und nicht nur am Ende alles löschen) – aber aufpassen, dass man nicht die falschen Objekte mit löscht (z.B. die ‘string list’ aller Files, die man bearbeiten will).

4a) Daten im Info Fenster ausgeben

Praat wird (sehr) langsam, wenn es etwas ins Info Fenster schreibt.

Also: Daten in ein File direkt abspeichern (z.B. mit ‘fileappend’).

4b) Daten im Info Fenster ausgeben

Zur Information des Users (und zum Testen, z.B. um Endlos-Schleifen zu erkennen) kann es sinnvoll sein, anzuzeigen, welches File grade bearbeitet wird. Bei der Bearbeitung sehr vieler Files bremsst das erheblich. Man sollte (nachdem man ein Skript mit wenigen Files getestet hat) nur am Ende des Skriptes einem ‘Fertig’ (oder so) ausdrucken, damit man weiß, dass das Skript am Ende angekommen ist. (Es kann etwas beunruhigend sein, wenn scheinbar ‘gar nichts’ geschieht, nachdem man ein Skript gestartet hat – ev. kontrollieren, ob das Ausgabefile größer wird.)

5) Praat Informationen ausgeben lassen

Praat informiert einen, wenn es etwas länger für etwas braucht (z.B. das Downsampeln bei der Formantberechnung oder Aktivität während der Pitchberechnung). Auch das bremsst das Skript. Man kann das unterdrücken, in dem man ‘noprogess’ vor das entsprechenden Kommando stellt, z.B.

```
noprogess To Pitch: 0, 75, 500
```