

Tutorial de Praat No. 2

Fernanda Barrientos Contreras

5 de julio de 2018

1. Scripting: Decirle a Praat que haga el trabajo

Supongamos que tienes 500 archivos de audio, cada uno con una palabra de 5 sonidos. Tenemos que etiquetar todo eso, medir los formantes de las vocales, y generar una tabla de valores para cada vocal.

Para mañana.

Parece una tarea de locos: anotar cada archivo, sacar un formante por vocal, y generar la tabla de valores para cada una. Quizás tenemos que pasar toda la noche despierto/as para lograrlo...

A menos que sepamos trabajar con scripts. Un script es un mini-programa que le dice a Praat algo como *Toma estos archivos, y haz en cada uno de ellos esto y esto*. Parece demasiado bueno para ser verdad. El problema: tenemos que programar un poco, pero solo con un poco de paciencia y buena vista se puede lograr. Veamos qué podemos hacer que nos sea útil para nuestro trabajo.

2. Cómo funciona un script?

Para que un script haga el trabajo, necesitamos:

- Definir elementos: por ejemplo, queremos decirle a Praat algo como *Abre todos los archivos con extensión .wav de mi carpeta de Descargas*. Para esto, necesitamos definir cuál es la ruta en tu computador hacia la carpeta de Descargas.
- Dar un comando: todo lenguaje de programación tiene una serie de comandos pre-determinados.

Una buena práctica es comentar los scripts: esto es, dejar una nota para nosotros mismos sobre lo que hace cada parte del script. En Praat esto se hace con el hash (`#`). Lo que queda en la línea después del hash es un comentario y Praat lo ignora cuando hacemos correr el script.

3. Abrir una serie de archivos

Aquí puedes encontrar un archivo .zip con una serie de 10 archivos .wav. Cada archivo es un audio que tiene una palabra de 5 fonos. Le diremos a Praat que los abra todos de una vez. Para esto bajaremos el archivo .zip, lo descomprimos, y ponemos la carpeta descomprimida en el Escritorio (Desktop).

Una vez que hemos hecho esto, vamos a Praat. En la ventana de Objetos vamos a *Praat* > *New Praat script*. Ahí pegamos el siguiente fragmento.

```
# Aquí definimos los directorios:
sound_directory$ = "/fernanda/Desktop/palabras"

# Esto crea una lista con los nombres de los archivos en tu carpeta.
Create Strings as file list... list 'sound_directory'/*.wav
noFiles = Get number of strings

# Esto abre los archivos que tengan los nombres de la lista.
for ifile to noFiles
select Strings list
filename$ = Get string... ifile
Read from file... 'sound_directory$/'filename$'
endfor
```

Donde dice `/fernanda/Desktop/palabras` reemplazamos con la ruta hacia tu carpeta. Si no la conoces, puedes hacer clic con el botón derecho del mouse sobre tu carpeta y luego hacer clic en Properties (Windows/Linux) o Get Info (Mac).

En las primeras líneas hemos definido los directorios (carpetas) con las que Praat va a trabajar. En la segunda parte hemos creado una lista con los nombres de los archivos en la carpeta. También creamos la variable `noFiles`, que es el número de archivos .wav en tu carpeta (es decir, 10). Finalmente, hemos creado lo que se llama en lenguaje de programación un *for loop*: es una orden donde el programa realiza la misma acción sobre una serie de objetos. Aquí le hemos dicho a Praat: *Mira la lista con los nombres de los archivos. Por cada nombre, busca el archivo en mi carpeta y ábrelo. Repite esta acción 10 veces.*

En el menú del script, vamos a *Run* > *Run*. Esto ejecutará el script que hemos creado. Para mayor seguridad, podemos guardar el script como un archivo. En el menú del script, vamos a *File* > *Save as...* Nos aseguramos de que la extensión del archivo sea `.praat`. Después podremos abrir este archivo en Praat y ejecutar todo esto de nuevo.

4. Generando TextGrids en forma automática

4.1. Crear una tabla de datos preliminares

Podemos generar fácilmente un Textgrid para cada archivo. Eso sí, tenemos que hacer las anotaciones nosotros/as mismo/as, pero al menos nos ahorramos una larga serie de clics (si por cada archivo necesitamos hacer 7 u 8 clics para crear el TextGrid, entonces repetir esto 10 veces = 70 u 80 clics ahorrados).

Sabemos entonces que cada archivo es una palabra con 5 fonos, por lo tanto podemos crear un TextGrid con 5 intervalos para cada uno. Lo que no sabemos con seguridad es la duración de cada archivo: si sabemos esto entonces podemos crear 5 intervalos que quepan al menos dentro del archivo de audio con la duración más corta.

Vamos a decirle a Praat entonces lo siguiente:

- *Crea una tabla donde pueda ver la duración total de cada archivo.*
- *Por cada archivo .wav que abriste desde mi carpeta, crea un TextGrid con dos tiers y uno de estos con 4 separaciones (boundaries) espaciados cada x ms.*

La primera operación requiere crear primero una tabla. Vamos a insertar en nuestro script, JUSTO DESPUÉS de la línea que dice `noFiles = Get number of strings`, lo siguiente (con el comentario también, para recordar lo que estamos haciendo):

```
# Esto crea una tabla llamada "datos", con 0 filas y dos columnas
# Una columna tiene el nombre del archivo, y la otra tiene la duración
table_ID = Create Table with column names... datos 0 Archivo Duracion
```

Como la operación se hace sobre los mismos archivos, podemos reutilizar el *for loop* que creamos arriba. Entonces, JUSTO ANTES de la línea que dice `endfor`, insertamos el siguiente fragmento:

```
# Esto llena la tabla con filas que tienen el nombre del archivo
# y la duración total de cada archivo.
end=Get total duration
select table_ID
Append row
Set string value... ifile Archivo 'filename$'
Set numeric value... ifile Duracion end
```

Ahora miramos la tabla. Cuál es el archivo más corto? Podemos ver que el de menor duración es `nandu.wav`, con 0.425 segundos.

4.2. Ahora sí: los TextGrids

Ahora sabemos que las cuatro separaciones de los TextGrids tienen que estar dentro del tiempo total del archivo más corto. Es razonable entonces poner cuatro separaciones cada 0.1 segundos (luego podemos mover las separaciones a mano, mientras anotamos). Le diremos entonces a Praat: *Por cada archivo, crea un TextGrid con un tier llamado fono. Luego crea una separación en 0.1, otra en 0.2, otra en 0.3, y otra en 0.4.*

Para lograr esto, necesitamos un comando para crear TextGrids. Entonces, JUSTO DESPUÉS de la línea `Read from file... 'sound_directory$'/'filename$'` insertamos lo siguiente:

```
# Esta parte crea un TextGrid con 4 separaciones por objeto Sound.
To TextGrid: "fono", ""
Insert boundary: 1, 0.1
Insert boundary: 1, 0.2
Insert boundary: 1, 0.3
Insert boundary: 1, 0.4
```

Esto creará un TextGrid por cada archivo de sonido; estos aparecerán en la ventana de Objetos. Ahora, anotamos manualmente y afinamos las separaciones de los fonos. Cuando estemos listos con la anotación, podemos guardar los TextGrids en nuestro computador. Para eso, **creamos el siguiente nuevo script**:

```
# Este script guarda los TextGrids en tu carpeta.
# Primero seleccionamos todo y le decimos a Praat:
# Todos los TextGrids en la selección tendrán un número de 1 a 10
select all
n = numberOfSelected ("TextGrid")
for i to n
tg[i] = selected ("TextGrid", i)
endfor
# Ahora le decimos a Praat: Toma un TextGrid y guárdalo como archivo
# con su nombre
for i to n
selectObject: tg[i]
tg$ = selected$("TextGrid")
Save as text file... 'tg$'.TextGrid
endfor
```

...Y corremos este pequeño script con *Run >Run*.

También podemos pegar este script al final del anterior, pero tenemos que estar muy atentos cuando lo hacemos correr: En vez de usar la opción *Run >Run*, tendremos que seleccionar esta parte y luego usar la opción *Run >Run selection*. Pero no es muy recomendado pues si estamos cansados se nos olvidará y terminaremos guardando TextGrids que están en blanco y habremos perdido el trabajo hecho (experiencia propia).

5. Obtener datos de un sonido en particular

Ahora que ya tenemos los sonidos anotados, queremos saber: ¿cuál es el valor del F1 en la primera vocal de la palabra *vivir* en *vivir.wav*? Para eso necesitamos extraer los sonidos de la palabra en cuestión, y extraer el formante.

5.1. Extraer sonidos automáticamente

Ahora haremos lo que hicimos en el Tutorial 1: extraer las vocales de cada una de estas palabras. Pero en lugar de hacerlo archivo por archivo, vamos a hacerlo con un script. Para hacer esto, necesitamos anotar los sonidos anteriores, de manera que todos tengan las vocales marcadas.

Creemos entonces un segundo script:

```
# Definimos nuestro directorio
source_directory$ = "/fernanda/Desktop/palabras"

# Esto crea una tabla llamada "datos2", con 5 columnas.
table_ID2= Create Table with column names... datos2 0 Vocal Duracion F1 F2
```

```

# Esto crea una lista con los nombres de los archivos .wav en tu carpeta.
wav_files = Create Strings as file list... wavlist 'source_directory$'/*.wav
nowav_files = Get number of strings

# Y esto crea una lista con los nombres de los archivos .TextGrid en tu carpeta.
tg_files = Create Strings as file list... tglis t 'source_directory$'/*.TextGrid
notg_files = Get number of strings

# Esto abre los archivos wav
for ifile to nowav_files
select Strings wavlist
wavfilename$ = Get string... ifile
wav_file= Read from file... 'source_directory$'/'wavfilename$'
palabra$= replace$ (wavfilename$, ".wav", "", 0)

# Y esto abre los archivos TextGrid
select Strings tglis t
tgfilename$ = Get string... ifile
tg_file= Read from file... 'source_directory$'/'tgfilename$'
endfor

```

Aquí hemos definido nuestro directorio (donde tenemos los sonidos y los TextGrid), y luego hemos dicho a Praat: *Crea una lista con los nombres de mis archivos.wav, otra lista con los archivos .TextGrid, y ábrelos*. Al mirar nuestra ventana de Objetos encontraremos los sonidos .wav y sus correspondientes TextGrids. En las primeras líneas hemos creado también una Tabla con 0 filas y 4 columnas, que luego llenaremos con los valores de los formantes.

Ahora extraemos las vocales de cada palabra. Para esto, pegamos el siguiente código JUSTO ANTES de la línea `endfor`:

```

# Esto extrae las vocales y elimina los audios originales.
select wav_file
plus tg_file
Extract intervals where... 1 no "matches (regex)" a|e|i|o|u
select wav_file
plus tg_file
Remove

```

Con esto le hemos dicho a Praat: *toma un sonido, toma su TextGrid, y extrae los intervalos que tengan una vocal a, e, i, o, u*. Luego le decimos a Praat que elimine el sonido con la palabra y su TextGrid de la ventana de Objetos (¡no te preocupes, están guardados en tu computador!), y deje solamente las vocales. Esto va a crear unos archivos de audio llamados: `Sound palabra_a_e_i_o_u_1`. El número final se refiere a la vocal en la palabra: por ejemplo, las vocales que corresponden a la palabra `pasto` se llamarán `Sound pasto_a_e_i_o_u_1` (es decir, la vocal `a`) y `Sound pasto_a_e_i_o_u_2`, es decir, la vocal `o`. En el próximo paso aprovecharemos de cambiar el nombre de estos archivos a algo más sencillo.

6. Extraer formantes

Ahora podemos también extraer formantes y sus respectivos valores en forma automática. Para esto crearemos un nuevo *for loop*: esta vez, le diremos a Praat *De todo lo que está en la ventana de Objetos, toma solo los que son de tipo Sound, y asígnales un número a cada uno*. Pegamos al final de nuestro script lo siguiente:

```
# Esto selecciona todos los sonidos
select all
minusObject: "Table datos2", "Strings wavlist", "Strings tglis"
n= numberOfSelected ("Sound")
for i from 1 to n
sound[i] = selected ("Sound", i)
endfor
```

Ahora vienen dos cosas importantes: cambiarle el nombre a los archivos, y conocer la duración de cada vocal. También definiremos el elemento *mid*: el punto medio de cada vocal. Praat simplemente tomará la duración total y la divide por 2.

```
# Esto cambia los nombres de los audios extraídos y calcula la duración
for i from 1 to n
selectObject: sound[i]
sound$=selected$("Sound")
nombre$=replace$ (sound$, "a_e_i_o_u_", "", 0)
Rename... 'nombre$'
end= Get total duration
mid= end/2
endfor
```

Ahora que tenemos cada sonido seleccionado, le decimos a Praat: *extrae los formantes F1 y F2 de cada sonido en el punto mid, considerando 5 formantes dentro de un máximo de 5,500 Hz*. Pegamos esto JUSTO ANTES de la línea *endfor*.

```
# Esto calcula los formantes F1 y F2.
To Formant (burg)... 0.001 5 5500 0.015 50
f1=Get value at time... 1 mid Hertz Linear
f2=Get value at time... 2 mid Hertz Linear
```

Pero ahora queremos registrar todos estos valores. Si recordamos bien, en 5.1 creamos una tabla llamada *datos2*, con 0 filas y con las columnas *Vocal*, *Duracion*, *F1*, *F2*. Ahora agregaremos los valores de cada vocal en la tabla. Le decimos entonces a Praat: *Toma los valores de cada vocal (duración, F1 y F2), y pégalos en la tabla a medida que analizas cada una*. Pegamos este código JUSTO ANTES de la línea *endfor*.

```
# Esto ingresa los valores en la tabla datos2
select table_ID2
Append row
Set string value... i Vocal 'nombre$'
Set numeric value... i Duracion end
Set numeric value... i F1 f1
Set numeric value... i F2 f2
endfor
```

Y finalmente, es buena práctica que nuestro script elimine todo lo que crea y deje la ventana de Objetos limpia. Hacemos eso, y dejamos solo la tabla, que además guardamos en nuestra carpeta. Pegamos esto DESPUÉS de la línea `endfor`.

```
# Con esto limpiamos la ventana de objetos y guardamos la tabla datos2
select all
minusObject: "Table datos2"
Remove
selectObject: "Table datos2"
Save as comma-separated file... /formant_table.csv
```

Y con esto ya tenemos gran parte del trabajo hecho.

7. Hacer más cosas con Praat

Si lo que deseas es, por ejemplo, medir el VOT de las obstruyentes, entonces tendrás que crear un tier adicional que marque los VOT. Para esto, en vez de medir formantes, podemos extraer los VOT y luego simplemente usar la línea `Get total duration` que hemos utilizado para saber la duración total de cada vocal. Para esto podemos olvidarnos de la medición de los formantes.

Si lo que quieres es averiguar la intensidad de cada sonido (esto es útil para cuando queremos investigar las aproximantes espirantes), entonces tendremos que modificar ligeramente el script que ya tenemos (esto es lo bueno de tener scripts: puedes modificarlos según las necesidades). Primero, tendremos que decirle a Praat que queremos extraer las aproximantes, y no las vocales. Para eso buscamos en el código la siguiente línea:

```
Extract intervals where... 1 no "matches (regex)" a|e|i|o|u
```

Que le dice a Praat *Extrae los intervalos en mi TextGrid que tengan escrito a, e, i, o, o u*. Modificamos entonces la última parte, la que dice `a|e|i|o|u`, y la cambiamos por `gf|bf`, que son las etiquetas que usamos en el TextGrid para anotar las aproximantes [ɣ] y [β] en los audios `trago.wav` y `vivir.wav`. La línea queda así:

```
# Esta línea extrae aproximantes en vez de vocales
Extract intervals where... 1 no "matches (regex)" gf|bf
```

Cuando hacemos correr el script (¡no aún!) con `Run>Run`, Praat nos mostrará el siguiente mensaje de error:

Pero esto no es un problema. Significa solamente que en la palabra que está analizando no encontró ningún intervalo en el TextGrid con estas etiquetas. Simplemente pulsamos OK y seguimos. Este mensaje aparecerá tantas veces como palabras sin las etiquetas especificadas haya (en nuestro caso, 8 veces).

Ahora nos ocupamos de medir la intensidad. Para esto, hacemos el siguiente cambio en el script. JUSTO DESPUÉS de la línea `mid=end/2`, agregar el cálculo de la intensidad total del sonido con la siguiente línea:

```
# Esta línea calcula la intensidad en dB de cada sonido
int= Get intensity (dB)
```

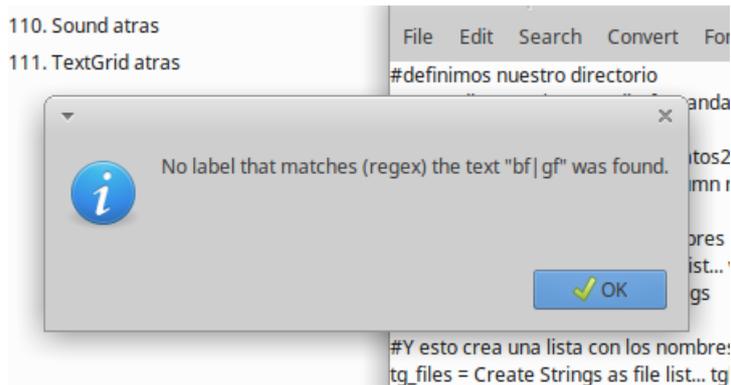


Figura 1: El error que no es un error. Simplemente presionar OK y seguir.

Y en la línea donde creamos la tabla de datos podemos agregar una columna para guardar la intensidad y que llamaremos dB. La línea entonces queda así:

```
table_ID2= Create Table with column names... datos2 0 Vocal Duracion F1 F2 dB
```

Y en la parte del script donde agregamos los valores calculados a la tabla agregamos el valor en dB. Ahí, JUSTO DEBAJO de la línea que dice:

```
Set numeric value... i F2 f2
```

Agregamos la siguiente línea:

```
# Esta línea agrega a la tabla la intensidad de cada sonido  
Set numeric value... i dB int
```

Luego corremos el script, y en la tabla `datos2` tendremos una columna adicional con la intensidad del sonido en dB.