

=====

Proyecto: Sistema de reconocimiento de habla en español - Gramatica Finita.
Archivo De Notas

=====

Alumno: Anastópulos Matías
Padrón: 95120

PARTE 1 - GRAMÁTICA FINITA.

Lo primero a realizar será especificar la gramática de nuestras frases. Se deberá crear en el directorio 'etc' el siguiente archivo con la especificación de la gramática:

```
$digit = cero | uno | dos | tres | cuatro | cinco | seis | siete | ocho | nueve;  
$name = juan | juana [fernandez] | patricia [rodriguez] | pedro | andrea [perez] | andres;  
( enviar-com ( llame|llamar) al <$digit> | ( (llame|llamar) a | comuniqueme con ) $name) enviar-fin)
```

Lo guardamos bajo el nombre 'gramatica'. Los símbolos que comienzan con '\$' indican símbolos no terminales, en este caso \$digit y \$name, que pueden expandirse a lo que hay del otro lado del '='. El símbolo '|' indica alternativas y los elementos entre llaves '[]' son opcionales.

PARTE 2 - RED DE PALABRAS.

Lo siguiente a realizar será crear la red de palabras que necesitará el comando 'HVite', que se realiza con el comando 'HParse'. Ejecutando parado en 'etc':

```
HParse gramatica wdnet.gf
```

Que genera la red 'wdnet.gf' a partir de la gramática en el archivo 'gramatica'.

PARTE 3 - DICCIONARIO DE PRONUNCIACIONES.

Lo siguiente a realizar será generar el diccionario de pronunciaciones. Parado en 'etc' me copio los dos archivos:

```
cp /home/cestien/proyecto/wlist.gf ./wlist.gf  
cp /home/cestien/proyecto/lexicon.gf ./lexicon.gf
```

Que son la lista de palabras y el diccionario fuente. Agregar la palabra 'al' alfabéticamente al 'wlist.gf' ejecutando parado en 'etc':

```
vim wlist.gf
```

Luego copiar de 'lexicon' (Diccionario del TP anterior) la línea de 'al':

```
al                                aa l
```

Agregar esa línea a 'lexicon.gf' alfabéticamente ejecutando parado en 'etc':

```
vim lexicon.gf
```

Generar el diccionario 'dict.gf' ejecutando parado en 'etc':

```
HDMan -m -w wlist.gf -l ../log/log.gf dict.gf lexicon.gf
```

Que busca en 'lexicon.gf' pronunciaciones de la lista 'wlist.gf' y genera el diccionario 'dict.gf'

PARTE 4 - GENERACIÓN DE FRASES.

Lo siguiente a realizar será generar la lista de las 200 frases a grabar para testeo. Esto se realiza con el comando 'HSGen'. Ejecutar parado en 'etc':

```
HSGen -l -n 200 wdnet.gf dict.gf > testprompts
```

Que genera la lista 'testpromps' con las frases. Será necesario eliminar los símbolos de comienzo y fin 'enviar-com' y 'enviar-fin'. Para esto ejecutar parado en 'etc' el siguiente comando:

```
awk '{ for(i=3;i<=NF-1;i++){printf "%s ", $i;} printf "\n"}' testprompts > testprompts2
```

Que genera el archivo testprompts2, que es igual al anterior, sin los símbolos de comienzo y fin. Luego se deberá incluir al comienzo de cada frase el nombre del archivo en que será grabado. Para eso ejecutar el siguiente comando parado en 'etc':

```
awk '{printf "Audio_%i          ",NR;  for(i=1;i<=NF;i++){printf "%s ", $i;} printf "\n"}' testprompts2 > testprompts3
```

Que genera el archivo testprompts3, que es similar a testprompts2, sólo que agrega al principio el nombre del archivo 'Audio_XXX'.

PARTE 5 - GRABADO DE AUDIOS.

Lo siguiente será grabar los audios. Se ha comenzado por grabar sólo los primeros 50 para una primera prueba. Se ha utilizado la aplicación de celular 'AudioRec' dado a que permite seleccionar la frecuencia de muestreo necesaria, que en este caso es 16k.

Una vez grabados se envían a la carpeta 'datos' con el siguiente comando:

```
scp -r /home/matias/Escritorio/wavgf manas@habla.fi.uba.ar:/home/manas/proyecto/datos
```

Que envía la carpeta 'wavgf' con los audios.

También será necesario crear el directorio a donde irán los coeficientes MFC.

```
mkdir mfcgf
```

PARTE 6 - GENERACIÓN DE COEFICIENTES MFC.

Lo siguiente será obtener los coeficientes MFC de los audios. Lo primero será crear el archivo con los archivos de origen y destino para 'HCopy'. Ejecutar en 'scripts':

```
nano listasgf
```

Y llenarlo con el siguiente contenido.

```
#!/bin/sh  
#Crea las listas archivos a procesar por HCopy  
#y los directorios en mfc, Ejecutarlo desde datos  
#  
  
if [ $# -ne 2 ]  
then  
echo Usar: $0 listatrain listatest  
exit 1  
fi  
  
ls ../datos/wavgf/* >p  
cat p |sed 's/wav/mfc/g' >q  
paste p q > $2  
rm p q
```

Lo que hará este script es generar la lista de archivos de origen y destino para el 'HCopy'. Luego le damos permiso de ejecución. Parado en 'scripts' ejecutar:

```
chmod +x listasgf
```

Luego lo ejecutamos parados en 'scripts':

```
./listasgf [] ../etc/gomfcgflistas
```

Que generará el erchivo 'gomfcgflistas' que tiene el siguiente contenido:

```
Archivo_Origen_1.wav      Archivo_Destino_1.mfc  
Archivo_Origen_2.wav      Archivo_Destino_2.mfc  
...
```

Lo siguiente será generar el archivo de configuración de 'HCopy'. Parado en 'config' ejecutar:

```
nano config.hcopygf
```

Para generar el archivo de configuración de 'HCopy', con el siguiente contenido:

```
# Coding parameters  
TARGETKIND = MFCC_0  
#TARGETKIND = MFCC_0_D_A  
TARGETRATE = 100000.0  
SAVECOMPRESSED = T
```

```
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
SOURCEFORMAT = WAV
```

Es similar al del trabajo anterior, solo que se cambia el archivo de origen a 'WAV'.

Finalmente ejecutamos el 'HCopy' para generar los coeficientes MFC. Ejecutar en 'datos':

```
HCopy -A -V -T 1 -C ../config/config.hcopygf -S ../etc/gomfcgflistas > ../log/hcopygf.log
```

PARTE 7 - RECONOCIMIENTO

Lo siguiente a realizar será generar el listado de archivos MFC. Para eso ejecutar en 'modelos':

```
ls ../datos/mfcgf/* > testgf.scp
```

Que generará 'testgf.scp', el listado de archivos con coeficientes mfc.

Lo siguiente será generar el Master Label File con las frases en 'testprompts3'. Parado en 'etc' ejecuto:

```
../scripts/prompts2mlf gfmlf testprompts3
```

Que generará el archivo 'gfmlf'.

Finalmente realizamos el reconocimiento ejecutando parados en 'rec':

```
HVite -C ../config/config -H ../modelos/hmm27/macros -H ../modelos/hmm27/hmmdefs -S ../modelos/testgf.scp -l '*' -i recoutgf1.mlf -w ../etc/wdnet.gf -p 0.0 -s 5.0 ../etc/dict.gf ../etc/monophones+sil3 &
```

Y verificamos los resultados ejecutando parados en 'etc':

```
HResults -t -f -I ../etc/gfmlf ../etc/wlist.gf recoutgf1.mlf
```

El resultado para 50 audios fue el siguiente:

```
----- Overall Results -----
SENT: %Correct=0.00 [H=0, S=50, N=50]
WORD: %Corr=89.35, Acc=53.26 [H=411, D=0, S=49, I=166, N=460]
=====
```

PARTE 8 - PRUEBA FINAL.

Finalmente se repite el proceso pero con los 200 audios grabados. El resultado para 200 audios fue el siguiente:

```
----- Overall Results -----
SENT: %Correct=0.00 [H=0, S=200, N=200]
WORD: %Corr=89.73, Acc=47.86 [H=1406, D=0, S=161, I=656, N=1567]
=====
```

```
ppelle@habla:/home/manas/proyecto$ cat etc/dict.gf
a          aa sp
al         aa l sp
andrea     aa n d rx ey aa sp
andres     aa n d rx ey s sp
cero       s ey rx ow sp
cinco      s iy n k ow sp
comunique  k ow m uw n iy k ey m ey sp
con        k ow n sp
cuatro     k w aa t rx ow sp
diego      d y ey gg ow sp
dos        d ow s sp
enviar-com [ ] sil sp
enviar-fin [ ] sil sp
fernandez  f ey rx n aa n d ey s sp
juan       hh w aa n sp
juana      hh w aa n aa sp
llamar     y aa m aa rx sp
llame      y aa m ey sp
nueve      n w ey bb ey sp
ocho       ow ch ow sp
patricia   p aa t rx iy s y aa sp
pedro      p ey dd rx ow sp
perez      p ey rx ey s sp
rodriguez  rr ow dd rx iy gg ey s sp
seis       s ey y s sp
siete      s y ey t ey sp
tres       t rx ey s sp
uno        uw n ow sp
```

} agregué esto y volví a hacer el HVite