

## Apéndice B

### Modelos Ocultos de Markov

#### Experimento I

##### *Archivo.cfg*

```
# lola edit
set config(edit,commandlist) {
}
set config(edit,input) ../labs/telel.train.phn
set config(edit,output) telel.train.map
set config(edit,script) ../script/remap.tcl

# lola map
set config(map,modelname) telel.0
set config(map,input) telel.train.map
set config(map,output) telel.train.mono

# feature extraction
set config(feature,basedir) C:/Download/tele-digi/
set config(feature,corpusdir) C:/Download/tele-digi/corpus
set config(feature,input) telel.train.mono
set config(feature,script) ../script/hfeat.tcl

# data selection
set config(data,modelname) telel.0
set config(data,maxwant) 200
set config(data,input) telel.train.mono
set config(data,output) telel.pick
set config(data,countfile) telel.count
set config(data,segments) 1

# hmm initialization
set config(init,pickfile) telel.pick
set config(init,numiter) 10
set config(init,basename) telel

# hmm training
set config(train,pickfile) telel.pick
set config(train,numiter) 10
set config(train,updatestate) 1
set config(train,updatetransp) 1
set config(train,basename) telel

# viterbi decoding
set config(search,wordlist) {
{cero {s e r o}}
{uno {u n o}}
```

```

{dos {dc d o s}}
{tres {tc t r e s}}
{cuatro {kc k w a tc t r o}}
{cinco {s i N kc k o}}
{seis {s e i s}}
{siete {s i e tc t e}}
{ocho {o tSc tS o}}
{nueve {n w e V e}}
{diez {dc d i e s}}
{.pau {.pau}}
}
set config(search,grammar) {
{telel {
  $telel = cero | uno | dos | tres | cuatro | cinco |
        seis | siete | ocho | nueve | diez;
$grammar = [.pau] < $telel [.pau] >;
}}
}
set config(search,searchfile) telel.search

set config(search,beam) 300.0
set config(search,wordendbeam) 100.0
set config(search,grammarscale) 10.0
set config(search,crossmodel) -30.0
set config(search,input) ../labs/telel.dev.wrd
set config(search,output) answer
set config(search,modelname) telel.1

# scoring
set config(score,suppress) { .pau
}
set config(score,reference) ../labs/telel.dev.wrd
set config(score,hypothesis) answer

set config(scribe,dictionary) ../dict/telel.db
set config(scribe,input) ../labs/telel.train.wrd
set config(scribe,output) telel.force
set config(scribe,word) 1
set config(scribe,phone) 1
set config(scribe,modelname) telel.1

set config(embed,trainfile) telel.force.mono
set config(embed,numiters) 8
set config(embed,prune) 1600.0
set config(embed,dumpstate) 1
set config(embed,basemodel) telel.1

```

## Archivo.desc

```

#!hscript
#
outputmodel "tele1.0";

vecsize 39;

prototype mono numstate 5 mixtures 4 transp
  0.000 1.000 0.000 0.000 0.000
  0.000 0.500 0.500 0.000 0.000
  0.000 0.000 0.500 0.500 0.000
  0.000 0.000 0.000 0.500 0.500
  0.000 0.000 0.000 0.000 0.000;

$back = o o_h o_x u u_h u_fp u_x w;
$dip = ae ai ei ia ie io oi wa we;
$flap = r rZ rr;
$fric = s s_v x f;
$front = e e_h e_fp e_x i i_h i_fp i_x;
$lat = l ;
$mid = a a_h a_fp a_x;
$nasa = m m_fp n N nj;
$noise = \.unk \.bn \.ln;
$pau = \.pau \.br;
$ssemi = D G V;
$uclosu = tc tSc pc kc;
$ustop = k p t tS;
$vclosu = dc dZc bc gc;
$vstop = b d dZ g;

define mono <x>;
define mono <j>;

define mono

$back<D $flap<D
  <D>
  D>$back D>$front D>$mid ;

define mono
$fric<G $front<G $mid<G $pau<G <G> G>$back G>$front
  G>$mid ;

define mono <N> ;

define mono $back<V $fric<V $front<V $lat<V $mid<V
  $noise<V $pau<V <V> V>$back V>$flap V>$front V>$lat
  V>$mid ;

```

```

define mono $back<a $dip<a $flap<a $fric<a $front<a $lat<a $mid<a
    $nasa<a $noise<a $pau<a $semi<a $ustop<a $vstop<a <a>
    a>$back a>$flap a>$fric a>$front a>$lat a>$mid
    ;

define mono $pau<a_fp $ustop<a_fp <a_fp>
    a_fp>$fric ;
define mono <a_h> ;
define mono <ae> ;
define mono <ai> ;

define mono b>$back b>$flap b>$front b>$lat b>$mid ;

define mono <bc> ;

define mono d>$back d>$flap d>$front d>$lat d>$mid ;

define mono dZ>$back dZ>$fric dZ>$front dZ>$mid ;

define mono <dZc> ;
define mono <dc> ;

define mono
$back<e $dip<e $flap<e $fric<e $front<e $lat<e $mid<e
    <e> e>$back
e>$flap e>$fric e>$front e>$lat
    e>$mid e>$nasa e>$noise e>$pau e>$semi e>$uclosu
    e>$vclosu e>$vstop ;
define mono $noise<e_fp $pau<e_fp $ustop<e_fp
    <e_fp> e_fp>$fric e_fp>$front e_fp>$nasa e_fp>$noise
    e_fp>$pau e_fp>$semi e_fp>$uclosu e_fp>$vclosu ;
define mono <e_h> ;
define mono <e_x> ;
define mono <ei> ;
define mono <f> ;

define mono g>$mid
    g>$back g>$flap g>$front ;

define mono <gc> ;
define mono $back<i $dip<i $flap<i $fric<i $front<i $lat<i $mid<i
    $nasa<i $noise<i $pau<i $semi<i $ustop<i
    $vstop<i <i> i>$back i>$dip i>$flap i>$fric i>$front i>$lat
    i>$mid i>$nasa i>$noise i>$pau i>$semi i>$uclosu i>$vclosu ;

```

```

define mono $noise<i_fp $pau<i_fp <i_fp>
    i_fp>$mid i_fp>$pau i_fp>$uclosu
    ;
define mono <i_h> ;
define mono <i_x> ;
define mono <ia> ;
define mono <ie> ;
define mono <io> ;

define mono k>$back k>$dip k>$flap k>$fric k>$front k>$lat k>$mid
k>$noise k>$pau k>$uclosu
    ;

define mono <kc> ;
define mono $back<l $dip<l $flap<l $fric<l $front<l $mid<l
    $nasa<l $noise<l $pau<l $semi<l $ustop<l
    $vstop<l <l> l>$back l>$dip l>$flap l>$fric l>$front
    l>$mid l>$nasa l>$noise l>$pau l>$semi l>$uclosu
    l>$vclosu ;

define mono <m> ;
define mono <m_fp> ;
define mono <n> ;
define mono <nj> ;
define mono $back<o $dip<o $flap<o $fric<o $front<o $lat<o $mid<o
    $nasa<o $noise<o $pau<o $semi<o $ustop<o
    $vstop<o <o> o>$back o>$flap o>$fric o>$front o>$lat
    o>$mid o>$nasa o>$noise o>$pau o>$semi o>$uclosu
    o>$vclosu ;

define mono <o_h> ;
define mono <o_x> ;
define mono <oi> ;
define mono p>$back p>$dip p>$flap p>$front p>$lat p>$mid ;

define mono <pc> ;
define mono <r> ;
define mono <rZ> ;
define mono <rr> ;
define mono <s> ;
define mono <s_v> ;
define mono t>$back t>$dip t>$flap t>$fric t>$front t>$lat t>$mid
    t>$nasa t>$noise t>$pau ;

define mono <tS> ;
define mono <tSc> ;
define mono <tc> ;
define mono $back<u $flap<u $fric<u $front<u $lat<u $mid<u
    $nasa<u $noise<u $pau<u $semi<u $ustop<u
    $vstop<u <u> u>$flap u>$fric u>$front u>$lat
    u>$mid u>$nasa u>$noise u>$pau u>$semi u>$uclosu
    u>$vclosu ;

```

```
define mono $mid<w $nasa<w <w> w>$front ;
define mono <we> ;
define mono $lat<a_fp ;
define mono $noise<a_fp ;
define mono a_fp>$flap ;
define mono a_fp>$front ;
define mono a_fp>$nasa ;
define mono a_fp>$noise ;
define mono $vstop<e_fp ;
define mono $front<i_fp ;
define mono <u_fp> ;
define mono u_fp>$nasa ;
define mono <u_x> ;
define mono <\\.ln> ;
define mono <\\.pau> ;
```

## Experimento II

Archivo .desc

```
#!hscript
#
outputmodel "tele1.0";

vecsize 39;

prototype mono numstate 5 mixtures 4 transp
  0.000 1.000 0.000 0.000 0.000
  0.000 0.500 0.500 0.000 0.000
  0.000 0.000 0.500 0.500 0.000
  0.000 0.000 0.000 0.500 0.500
  0.000 0.000 0.000 0.000 0.000;

define mono
<a>
<a_h>
<a_fp>
<b>
<bc>
<d>
<D>
<dc>
<e>
<e_h>
<e_fp>
<e_x>
<f>
<g>
<G>
<gc>
<i>
<i_h>
<i_fp>
<i_x>
<j>
<x>
<k>
<kc>
<l>
<dZ>
<dZc>
<m>
<m_fp>
<n>
<N>
<nj>
<o>
<o_h>
<o_x>
<p>
<pc>
<r>
```

<rr>  
<rZ>  
<s>  
<s\_v>  
<t>  
<tc>  
<tS>  
<tSc>  
<u>  
<u\_fp>  
<u\_x>  
<w>  
<V>  
<ae>  
<ai>  
<ei>  
<ia>  
<ie>  
<io>  
<oi>  
<wa>  
<we>  
<.bn>  
<.br>  
<.ln>  
<.unk>  
<.pau>;



**Experimento III**

Archivo.desc

```

#!hscript
#
outputmodel "tele1.0";

vecsize 39;

prototype mono numstate 5 mixtures 4 transp
0.000 1.000 0.000 0.000 0.000
0.000 0.500 0.500 0.000 0.000
0.000 0.000 0.500 0.500 0.000
0.000 0.000 0.000 0.500 0.500
0.000 0.000 0.000 0.000 0.000;

$back = o o_h o_x u u_h u_fp u_x w;
$dip = ae ai ei ia ie io oi wa we;
$flap = r rZ rr;
$fric = s s_v x f;
$front = e e_h e_fp e_x i i_h i_fp i_x;
$lat = l ;
$mid = a a_h a_fp a_x;
$nasa = m m_fp n N nj;
$noise = \.unk \.bn \.ln;
$pau = \.pau \.br;
$semi = D G V;
$uclosu = tc tSc pc kc;
$ustop = k p t tS;
$vclosu = dc dZc bc gc;
$vstop = b d dZ g;

define mono <x>;
define mono <j>;

define mono

$back<D $flap<D
<D>
D>$back D>$front D>$mid ;

define mono
$fric<G $front<G $mid<G $pau<G <G> G>$back G>$front
G>$mid ;

define mono <N> ;

define mono $back<V $fric<V $front<V $lat<V $mid<V
$noise<V $pau<V <V> V>$back V>$flap V>$front V>$lat

```

```

V>$mid ;

define mono $back<a $dip<a $flap<a $fric<a $front<a $lat<a $mid<a
  $nasa<a $noise<a $pau<a $semi<a $ustop<a $vstop<a <a>
  a>$back a>$flap a>$fric a>$front a>$lat a>$mid
  ;

define mono $pau<a_fp $ustop<a_fp <a_fp>
  a_fp>$fric ;
define mono <a_h> ;
define mono <ae> ;
define mono <ai> ;

define mono b>$back b>$flap b>$front b>$lat b>$mid ;

define mono <bc> ;

define mono d>$back d>$flap d>$front d>$lat d>$mid ;

define mono dZ>$back dZ>$fric dZ>$front dZ>$mid ;

define mono <dZc> ;
define mono <dc> ;

define mono
$back<e $dip<e $flap<e $fric<e $front<e $lat<e $mid<e
  <e> e>$back
e>$flap e>$fric e>$front e>$lat
  e>$mid e>$nasa e>$noise e>$pau e>$semi e>$uclosu
  e>$vclosu e>$vstop ;
define mono $noise<e_fp $pau<e_fp $ustop<e_fp
  <e_fp> e_fp>$fric e_fp>$front e_fp>$nasa e_fp>$noise
  e_fp>$pau e_fp>$semi e_fp>$uclosu e_fp>$vclosu ;
define mono <e_h> ;
define mono <e_x> ;
define mono <ei> ;
define mono <f> ;

define mono g>$mid
  g>$back g>$flap g>$front ;

define mono <gc> ;
define mono $back<i $dip<i $flap<i $fric<i $front<i $lat<i $mid<i
  $nasa<i $noise<i $pau<i $semi<i $ustop<i
  $vstop<i <i> i>$back i>$dip i>$flap i>$fric i>$front i>$lat

```

```

i>$mid i>$nasa i>$noise i>$pau i>$semi i>$uclosu i>$vclosu ;

define mono $noise<i_fp $pau<i_fp <i_fp>
  i_fp>$mid i_fp>$pau i_fp>$uclosu
;
define mono <i_h> ;
define mono <i_x> ;
define mono <ia> ;
define mono <ie> ;
define mono <io> ;

define mono k>$back k>$dip k>$flap k>$fric k>$front k>$lat k>$mid
k>$noise k>$pau k>$uclosu
;

define mono <kc> ;
define mono $back<l $dip<l $flap<l $fric<l $front<l $mid<l
  $nasa<l $noise<l $pau<l $semi<l $ustop<l
  $vstop<l <l> l>$back l>$dip l>$flap l>$fric l>$front
  l>$mid l>$nasa l>$noise l>$pau l>$semi l>$uclosu
  l>$vclosu ;

define mono <m> ;
define mono <m_fp> ;
define mono <n> ;
define mono <nj> ;
define mono $back<o $dip<o $flap<o $fric<o $front<o $lat<o $mid<o
  $nasa<o $noise<o $pau<o $semi<o $ustop<o
  $vstop<o <o> o>$back o>$flap o>$fric o>$front o>$lat
  o>$mid o>$nasa o>$noise o>$pau o>$semi o>$uclosu
  o>$vclosu ;

define mono <o_h> ;
define mono <o_x> ;
define mono <oi> ;
define mono p>$back p>$dip p>$flap p>$front p>$lat p>$mid ;

define mono <pc> ;
define mono <r> ;
define mono <rZ> ;
define mono <rr> ;
define mono <s> ;
define mono <s_v> ;
define mono t>$back t>$dip t>$flap t>$fric t>$front t>$lat t>$mid
  t>$nasa t>$noise t>$pau ;

define mono <tS> ;
define mono <tSc> ;
define mono <tc> ;
define mono $back<u $flap<u $fric<u $front<u $lat<u $mid<u
  $nasa<u $noise<u $pau<u $semi<u $ustop<u
  $vstop<u <u> u>$flap u>$fric u>$front u>$lat
  u>$mid u>$nasa u>$noise u>$pau u>$semi u>$uclosu

```

```
u>$vclosu ;
```

```
define mono $mid<w $nasa<w <w> w>$front ;  
define mono <we> ;  
define mono $lat<a_fp ;  
define mono $noise<a_fp ;  
define mono a_fp>$flap ;  
define mono a_fp>$front ;  
define mono a_fp>$nasa ;  
define mono a_fp>$noise ;  
define mono $vstop<e_fp ;  
define mono $front<i_fp ;  
define mono <u_fp> ;  
define mono u_fp>$nasa ;  
define mono <u_x> ;  
define mono <\\.ln> ;  
define mono <\\.pau> ;
```