

Recuperación de Información en Textos Hablados

Jorge Luis Guevara Díaz

Sección de Postgrado en Ciencias Físicas y Matemáticas
Maestría en Ciencia de la Computación
Universidad Nacional de Trujillo



26 de enero de 2011

Contenido

1 Introducción

- Recuperación de Información en Texto Hablados
- Reconocimiento Automático del Habla
- Objetivo

2 Algoritmo

- Descripción del Algoritmo
- Preparación de los Datos
- Construcción de los HMM
- Creación de fonemas dependientes del contexto
- Decodificación
- Algoritmo Propuesto

3 Resultados

- Resultados por Sentencias
- Resultados variando parámetros: Penalidad por inserción y factor de escala
- Evaluación del Algoritmo usando k-fold
- Prueba de Significancia Estadística de t-student

4 Conclusiones

5 Referencias



Recuperación de Información en Texto Hablados

Recuperación de Información en Texto Hablados

Es la **transcripción** de archivos con habla hacia una secuencia de palabras usando un computador. En este contexto:

- **Recuperación** significa transcripción de los archivos de audio.



Recuperación de Información en Texto Hablados

Recuperación de Información en Texto Hablados

Es la **transcripción** de archivos con habla hacia una secuencia de palabras usando un computador. En este contexto:

- **Recuperación** significa transcripción de los archivos de audio.
- **Información** significa secuencia organizada de caracteres en palabras con algún significado.



Recuperación de Información en Texto Hablados

Recuperación de Información en Texto Hablados

Es la **transcripción** de archivos con habla hacia una secuencia de palabras usando un computador. En este contexto:

- **Recuperación** significa transcripción de los archivos de audio.
- **Información** significa secuencia organizada de caracteres en palabras con algún significado.
- **Textos hablados** son grabaciones de audio con habla realizada por una persona en particular.



Recuperación de Información en Texto Hablados

Recuperación de Información en Texto Hablados

Es la **transcripción** de archivos con habla hacia una secuencia de palabras usando un computador. En este contexto:

- **Recuperación** significa transcripción de los archivos de audio.
- **Información** significa secuencia organizada de caracteres en palabras con algún significado.
- **Textos hablados** son grabaciones de audio con habla realizada por una persona en particular.



Reconocimiento Automático del Habla

Reconocimiento Automático del Habla

$$\mathbf{W} = \operatorname{argmax}_w P(\mathbf{W}|\mathbf{O}) = \operatorname{argmax}_w \frac{P(\mathbf{W})P(\mathbf{O}|\mathbf{W})}{P(\mathbf{O})} \quad (1)$$

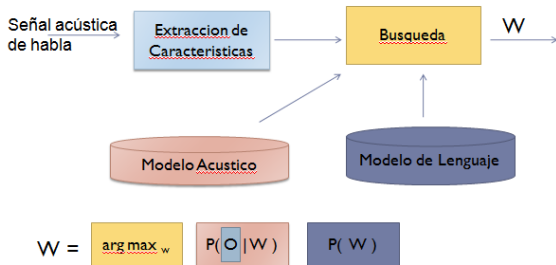


Figura: Arquitectura del Reconocimiento Automático del Habla



Objetivo

Objetivo

Proponer un **algoritmo para recuperar información de textos hablados dependientes del hablante con información a priori del lenguaje**, haciendo uso de conceptos de reconocimiento automático del habla para palabras continuas.

El algoritmo propuesto se basa en técnicas del reconocimiento automático del habla continua dependiente del hablante



Objetivo

Objetivo

Proponer un **algoritmo para recuperar información de textos hablados dependientes del hablante con información a priori del lenguaje**, haciendo uso de conceptos de reconocimiento automático del habla para palabras continuas.

El algoritmo propuesto se basa en técnicas del reconocimiento automático del habla continua dependiente del hablante



Descripción del Algoritmo

Descripción del Algoritmo

1 Preparación de los datos

- Diseño de la gramática
- Construcción del diccionario de pronunciación
- Obtención de los datos
- Construcción de los archivos de transcripción
- Extracción de características usando MFCC

2 Construcción de los HMM

- Definición de la estructura del HMM
- Estimación de los parámetros iniciales del HMM
- Entrenamiento de los HMM
- Establecer HMM para silencio y pausa



Descripción del Algoritmo

Descripción del Algoritmo

1 Preparación de los datos

- Diseño de la gramática
- Construcción del diccionario de pronunciación
- Obtención de los datos
- Construcción de los archivos de transcripción
- Extracción de características usando MFCC

2 Construcción de los HMM

- Definición de la estructura del HMM
- Estimación de los parámetros iniciales del HMM
- Entrenamiento de los HMM
- Establecer HMM para silencio y pausa

3 Creación de fonemas dependientes del contexto

- Creación de los trifenemas
- Clustering de estados



Descripción del Algoritmo

Descripción del Algoritmo

1 Preparación de los datos

- Diseño de la gramática
- Construcción del diccionario de pronunciación
- Obtención de los datos
- Construcción de los archivos de transcripción
- Extracción de características usando MFCC

2 Construcción de los HMM

- Definición de la estructura del HMM
- Estimación de los parámetros iniciales del HMM
- Entrenamiento de los HMM
- Establecer HMM para silencio y pausa

3 Creación de fonemas dependientes del contexto

- Creación de los trifenemas
- Clustering de estados

4 Decodificación

- Algoritmo Token-passing



Descripción del Algoritmo

Descripción del Algoritmo

1 Preparación de los datos

- Diseño de la gramática
- Construcción del diccionario de pronunciación
- Obtención de los datos
- Construcción de los archivos de transcripción
- Extracción de características usando MFCC

2 Construcción de los HMM

- Definición de la estructura del HMM
- Estimación de los parámetros iniciales del HMM
- Entrenamiento de los HMM
- Establecer HMM para silencio y pausa

3 Creación de fonemas dependientes del contexto

- Creación de los trifenemas
- Clustering de estados

4 Decodificación

- Algoritmo Token-passing



Descripción del Algoritmo

Descripción del Algoritmo

1 Preparación de los datos

- Diseño de la gramática
- Construcción del diccionario de pronunciación
- Obtención de los datos
- Construcción de los archivos de transcripción
- Extracción de características usando MFCC

2 Construcción de los HMM

- Definición de la estructura del HMM
- Estimación de los parámetros iniciales del HMM
- Entrenamiento de los HMM
- Establecer HMM para silencio y pausa

3 Creación de fonemas dependientes del contexto

- Creación de los trifenemas
- Clustering de estados

4 Decodificación

- Algoritmo Token-passing



Gramática libre del contexto

Gramática Libre del Contexto

Es una 4-tupla $G = (N, T, P, S)$, con N y T son los conjuntos de símbolos terminales y no terminales donde $N \cap T = \phi$.

P es el conjunto de reglas de la forma $A \leftarrow \alpha$, donde $A \in N$, $\alpha \in (N \cup T)^*$ y $S \in T$ es el símbolo inicial.

Ejemplo de GLC

$S \rightarrow (\text{SENT-START}(\text{TELEFONO } \langle \$\text{digit} \rangle | (\text{LLAMAR} | \text{MARCAR}) \$\text{name}) \text{SENT-END})$

$\$ \text{digit} \rightarrow \text{UNO} | \text{DOS} | \text{TRES} | \text{CUATRO} | \text{CINCO}$
 $| \text{SEIS} | \text{SIETE} | \text{OCHO} | \text{NUEVE} | \text{CERO};$

$\$ \text{name} \rightarrow [\text{JORGE}] \text{LUIS} |$
 $[\text{LEISSI}] \text{LEON} |$
 $[\text{CARLOS}] \text{DIAZ} |$
 $[\text{LUIS}] \text{GUEVARA};$



Gramática libre del contexto

Gramática Libre del Contexto

Es una 4-tupla $G = (N, T, P, S)$, con N y T son los conjuntos de símbolos terminales y no terminales donde $N \cap T = \phi$.

P es el conjunto de reglas de la forma $A \leftarrow \alpha$, donde $A \in N$, $\alpha \in (N \cup T)^*$ y $S \in T$ es el símbolo inicial.

Ejemplo de GLC

$S \rightarrow (\text{SENT-START}(\text{TELEFONO } \langle \$\text{digit} \rangle | (\text{LLAMAR} | \text{MARCAR}) \$\text{name}) \text{SENT-END})$

$\$ \text{digit} \rightarrow \text{UNO} | \text{DOS} | \text{TRES} | \text{CUATRO} | \text{CINCO}$
 $| \text{SEIS} | \text{SIETE} | \text{OCHO} | \text{NUEVE} | \text{CERO};$

$\$ \text{name} \rightarrow [\text{JORGE}] \text{LUIS} |$
 $[\text{LEISSI}] \text{LEON} |$
 $[\text{CARLOS}] \text{DIAZ} |$
 $[\text{LUIS}] \text{GUEVARA};$



Gramática libre del contexto

Grafo de Palabras

Grafo $G = (V(G), E(G), \psi_G)$ generado por una GLC. El conjunto de vértices $V(G)$ representa el conjunto de palabras. EL conjunto de aristas $E(G)$ y la función de incidencia ψ_G modelan la transición entre palabras

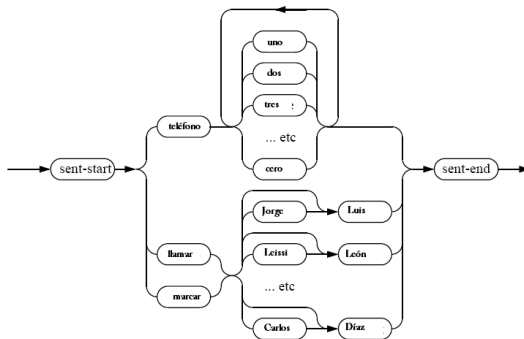


Figura: Grafo de palabras correspondiente a la gramática anterior.



Algoritmo procesamiento gramática

ALGORITMO-PROCESAMIENTO-GRAMÁTICA (GRAMÁTICA GLC G)

▷ Construcción de los nodos

▷ Para cada símbolo de G

1 $i \leftarrow 0$

2 **for** cada X en G

do

3 **if** X =símbolo terminal

then

 ▷ Crear Nodo

4 $I_i \leftarrow \text{Crear-Nodo}(X, \text{Palabra})$

5 $i \leftarrow i + 1$

▷ Construcción de las aristas

6 **for** $i \leftarrow 0$ **to** N

do

7 **for** $j \leftarrow 0$ **to** N

do

8 **if** $\text{Palabra}(I_i)$ está antes de $\text{Palabra}(I_j)$ en G

then

9 $J_{i,j} \leftarrow \text{Crear-Arista}(I_i, I_j)$



Construcción del Diccionario de Pronunciación

CONSTRUCCIÓN-DICCIONARIO-PRONUNCIACIÓN (DICCIONARIO D , GRAMÁTICA GLC G)

```
    ▷ Conjunto de Fonemas
1   $S \leftarrow \phi$ 
    ▷ Para cada símbolo terminal de  $G$ 
2  for  $a$  en  $G$ 
      do
3      ▷ Extraer transcripción fonética de la palabra
4       $t \leftarrow \text{TranscripciónFonética}(D,a)$ 
5      ▷ agregar nuevos fonemas encontrados
6       $S \leftarrow S \cup t$ 
7      ▷ escribir
8       $DIC \leftarrow \text{write}(a,t)$ 

9  return  $DIC$ 
```



Construcción del Diccionario de Pronunciación

CARLOS	k ah r l oh s sp
CERO	th eh r oh sp
CINCO	th ih ng k oh sp
CUATRO	k w ah t r oh sp
DIAZ	dh ih ah s sp
DOS	dh oh s sp
GUEVARA	g eh b ah r ah sp
JORGE	j oh r j eh sp
.	
.	
.	
SIETE	s y eh t eh sp
silencio	sil
TELEFONO	t eh l eh f oh n oh sp
TRES	t r eh s sp
UNO	uh n oh sp

Los fonemas *sp* y *sil* denotan silencio.



Construcción del Diccionario de Pronunciación

Lista de fonemas formato IPA

k
ah
r
l
oh
s
sp
.
.
.
sil
f



Generación aleatoria de Sentencias

GENERACIÓN-ALEATORIA-SENTENCIAS (GRAMÁTICA GLC G)

```
    ▷  $S_0$  es el símbolo inicial
1  Pila.push( $S_0$ )
2  while (Not)Pila.vacia
    do
3       $s \leftarrow Pila.pop()$ 
4      if  $s$  es terminal
        then
5          sentencia.add( $s$ )
6          print  $s$ 
7      else
8          ▷ Escoger una regla de manera aleatoria para  $s$ 
9           $p = s \rightarrow \alpha$ 
10         Pila.push(reverso( $\alpha$ ))
11
12
13 return sentencia
```

[4] Algoritmo de Purdom



Generación aleatoria de Sentencias

1. TELEFONO DOS SEIS CUATRO TRES UNO OCHO DOS CINCO CINCO CUATRO UNO SEIS CUATRO
SEIS CERO CINCO SIETE CUATRO NUEVE CERO
2. LLAMAR LUIS GUEVARA
3. TELEFONO SEIS SIETE SIETE UNO UNO DOS DOS UNO SIETE CINCO CUATRO OCHO
4. LLAMAR DIAZ
5. LLAMAR GUEVARA
6. MARCAR LEON
7. MARCAR LUIS GUEVARA
8. TELEFONO OCHO SIETE SIETE DOS CERO TRES OCHO CERO CUATRO
9. TELEFONO UNO
10. LLAMAR LEON
11. LLAMAR LUIS
12. TELEFONO SEIS TRES
13. LLAMAR LEISSI LEON
14. TELEFONO DOS OCHO NUEVE SIETE CUATRO CINCO CERO DOS CERO TRES CINCO CINCO SEIS
NUEVE NUEVE CINCO OCHO DOS
- .
- .
- .
197. MARCAR CARLOS DIAZ
198. TELEFONO CINCO CERO OCHO SEIS SEIS CERO DOS UNO CERO CINCO OCHO CUATRO TRES SEIS
CINCO SIETE TRES DOS CINCO CUATRO CERO
199. LLAMAR LEON
200. TELEFONO SEIS CUATRO CERO DOS UNO



Construcción de los Archivos de Transcripción

ALGORITMO-CONSTRUCCIÓN-ARCHIVOS-TRANSCRIPCIÓN (SENTENCIA S , DICCIONARIO D)

▷ Conjunto de Palabras

1 $P \leftarrow \phi$

▷ Para cada palabra w en la sentencia S

2 **for** w en S

3 **do**

4 $P \leftarrow P \cup w$

5

6 archivoTranscripción \leftarrow **write**(P , extraerFonema(P, D))

7 **return** archivoTranscripción



Construcción de los Archivos de Transcripción

"*/T0001.lab"	.	DIAZ	OCHO
TELEFONO	"*/T0003.lab"	.	CUATRO
DOS	TELEFONO	"*/T0195.lab"	TRES
SEIS	SEIS	TELEFONO	SEIS
CUATRO	SIETE	CERO	CINCO
TRES	SIETE	.	SIETE
UNO	UNO	"*/T0196.lab"	TRES
OCHO	UNO	MARCAR	DOS
DOS	DOS	GUEVARA	CINCO
CINCO	DOS	.	CUATRO
CINCO	UNO	"*/T0197.lab"	CERO
CUATRO	SIETE	MARCAR	.
UNO	CINCO	CARLOS	"*/T0199.lab"
SEIS	CUATRO	DIAZ	LLAMAR
CUATRO	OCHO	.	LEON
SEIS	.	"*/T0198.lab"	.
CERO	.	TELEFONO	"*/T0200.lab"
CINCO	.	CINCO	TELEFONO
SIETE	"*/T0193.lab"	CERO	SEIS
CUATRO	TELEFONO	OCHO	CUATRO
NUEVE	DOS	SEIS	CERO
CERO	TRES	SEIS	DOS
.	SIETE	CERO	UNO
"*/T0002.lab"	.	DOS	.



Construcción de los Archivos de Transcripción a Nivel Fonema

"T0001.lab"	dh	eh	.	s	ng	l
sil	oh	y	"T0002.lab"	s	k	eh
t	s	s	sil	y	oh	f
eh	th	th	ll	eh	k	oh
l	ih	eh	ah	t	w	n
eh	ng	r	m	eh	ah	oh
f	k	oh	ah	s	t	s
oh	oh	th	r	y	r	eh
n	th	ih	l	eh	oh	y
oh	ih	ng	uh	t	oh	s
dh	ng	k	y	eh	ch	k
oh	k	oh	s	uh	oh	w
s	oh	s	g	n	sil	ah
s	k	y	eh	oh	.	t
eh	w	eh	b	uh	.	r
y	ah	t	ah	n	.	oh
s	t	eh	r	oh	"T0199.lab"	th
k	r	k	ah	dh	sil	eh
w	oh	w	sil	oh	ll	r
ah	uh	ah	.	s	ah	oh
t	n	t	"T0003.lab"	dh	m	dh
r	oh	r	sil	oh	ah	oh
oh	s	oh	t	s	r	s
t	eh	n	eh	uh	l	uh
r	y	w	l	n	eh	n
eh	s	eh	eh	oh	oh	oh
s	k	b	f	s	n	sil
uh	w	eh	oh	y	sil	.
n	ah	th	n	eh	.	
oh	t	eh	oh	t	"T0200.lab"	
oh	r	r	s	eh	sil	
ch	oh	oh	eh	th	t	
oh	s	sil	y	ih	eh	



Extracción de Características con MFCC

MFCC

Algoritmo Coeficientes Cepstrales en Frecuencia Mel propuesto por [2]

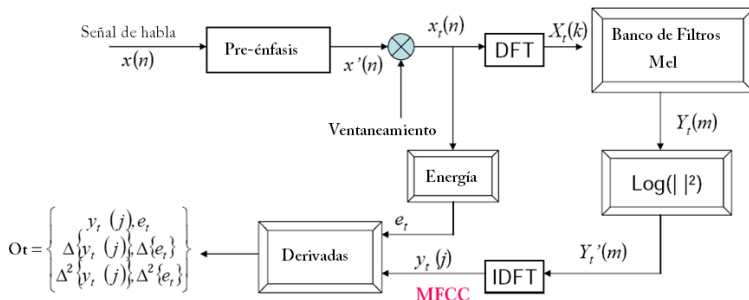


Figura: Proceso de construcción de los MFCC.



Extracción de Características con MFCC

- Digitalización y cuantificación de la señal.

$$x[n] = x_a(n\Delta T) \quad (2)$$

- Pre-énfasis en altas frecuencias.

$$y[n] = x[n] + \alpha x[n - 1] \quad (3)$$



Extracción de Características con MFCC

- Digitalización y cuantificación de la señal.

$$x[n] = x_a(n\Delta T) \quad (2)$$

- Pre-énfasis en altas frecuencias.

$$y[n] = x[n] + \alpha x[n - 1] \quad (3)$$

- Ventaneamiento.

$$x_s[n] = y[n]h_{h_s}[n] \quad (4)$$



Extracción de Características con MFCC

- Digitalización y cuantificación de la señal.

$$x[n] = x_a(n\Delta T) \quad (2)$$

- Pre-énfasis en altas frecuencias.

$$y[n] = x[n] + \alpha x[n - 1] \quad (3)$$

- Ventaneamiento.

$$x_s[n] = y[n]h_{h_s}[n] \quad (4)$$

- Análisis con Transformada de Fourier.

$$X_s(\omega) = \sum_{n=1}^{\infty} x_s[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} h_h[s - n]x[n]e^{-j\omega n}$$



Extracción de Características con MFCC

- Digitalización y cuantificación de la señal.

$$x[n] = x_a(n\Delta T) \quad (2)$$

- Pre-énfasis en altas frecuencias.

$$y[n] = x[n] + \alpha x[n - 1] \quad (3)$$

- Ventaneamiento.

$$x_s[n] = y[n]h_{h_s}[n] \quad (4)$$

- Análisis con Transformada de Fourier.

$$X_s(\omega) = \sum_{n=1}^{\infty} x_s[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} h_h[s - n]x[n]e^{-j\omega n} \quad (5)$$



Extracción de Características con MFCC

- Banco de Filtros Mel.

$$H_m[k] = \begin{cases} 0 & \text{si } k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{k-f(m-1)}{f(m+1)-f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \quad (6)$$

donde

$$f(m) = \frac{N}{F_s} \beta^{-1}(\beta(f_1) + m \frac{\beta(f_h) - \beta(f_1)}{M+1}) \quad (7)$$

La escala Mel β esta dada por:

$$\beta(f) = 1125 \ln(1 + \frac{f}{700}) \quad (8)$$

y su inversa β^{-1} esta dada por:

$$\beta^{-1}[b] = 700(\exp(\frac{b}{1125}) - 1) \quad (9)$$

Luego se computa el logaritmo de la energía de cada filtro:

$$S(m) = \ln(\sum_{k=0}^{N-1} |X_a(k)|^2 H_m(k)), \quad 0 < m < M \quad (10)$$



Extracción de Características con MFCC

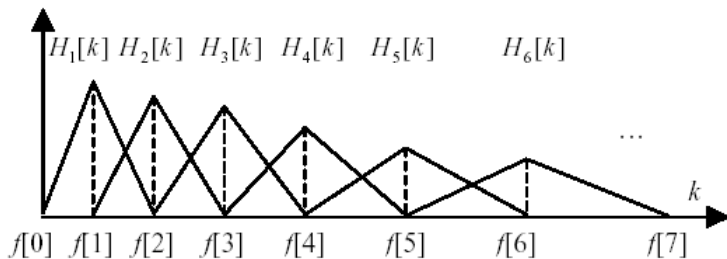


Figura: Banco de filtros en escala mel .



Extracción de Características con MFCC

- Análisis Cepstral.

$$c(m) = \sum_{n=0}^{M-1} S(n) \cos(\pi n (\frac{m - \frac{1}{2}}{M})) \quad (11)$$

- Coeficientes Cepstrales Dinámicos.

$$\begin{bmatrix} c_k \\ \Delta c_k \\ \Delta\Delta c_k \\ E \\ \Delta E \\ \Delta\Delta E \end{bmatrix} \quad (12)$$

Donde

$$E = |x_s[n]|^2$$



Extracción de Características con MFCC

- Análisis Cepstral.

$$c(m) = \sum_{n=0}^{M-1} S(n) \cos(\pi n (\frac{m - \frac{1}{2}}{M})) \quad (11)$$

- Coeficientes Cepstrales Dinámicos.

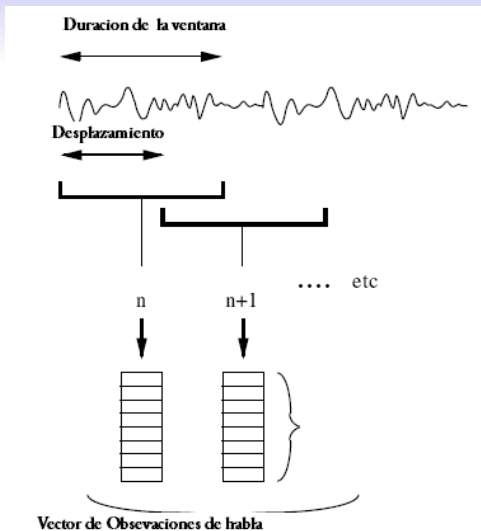
$$\begin{bmatrix} c_k \\ \Delta c_k \\ \Delta\Delta c_k \\ E \\ \Delta E \\ \Delta\Delta E \end{bmatrix} \quad (12)$$

Donde

$$E = |x_s[n]|^2$$



Extracción de Características con MFCC



Descripción del Algoritmo

Descripción del Algoritmo

1 Preparación de los datos

- Diseño de la gramática
- Construcción del diccionario de pronunciación
- Obtención de los datos
- Construcción de los archivos de transcripción
- Extracción de características usando MFCC

2 Construcción de los HMM

- Definición de la estructura del HMM
- Estimación de los parámetros iniciales del HMM
- Entrenamiento de los HMM
- Establecer HMM para silencio y pausa

3 Creación de fonemas dependientes del contexto

- Creación de los trifenemas
- Clustering de estados

4 Decodificación

- Algoritmo Token-passing



Modelos Ocultos de Markov HMM

HMM

Extensión de las cadenas de Markov que incorporan un proceso oculto en su modelado [3], [1], [5]

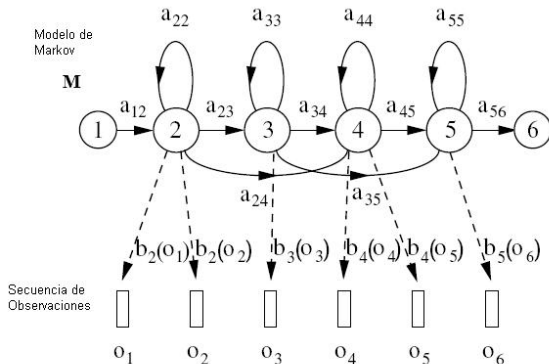


Figura: Modelo Oculto de Markov.



Modelos Ocultos de Markov HMM

- 1 **Evaluación**, Dada una secuencia de observaciones $\vec{o}_1 \vec{o}_2 \dots \vec{o}_T$ y un HMM $\lambda(A, B, \pi)$, como calcular eficientemente $P(O|\lambda)$.
- 2 **Decodificación** Dada una secuencia de observaciones $\vec{o}_1 \vec{o}_2 \dots \vec{o}_T$ y un HMM $\lambda(A, B, \pi)$, como escoger una correspondiente secuencia de estados $Q = q_1 q_2 \dots q_T$, que sea óptima en el sentido de que mejor explique la observación generada.



Modelos Ocultos de Markov HMM

- 1 **Evaluación**, Dada una secuencia de observaciones $\vec{o}_1 \vec{o}_2 \dots \vec{o}_T$ y un HMM $\lambda(A, B, \pi)$, como calcular eficientemente $P(O|\lambda)$.
- 2 **Decodificación** Dada una secuencia de observaciones $\vec{o}_1 \vec{o}_2 \dots \vec{o}_T$ y un HMM $\lambda(A, B, \pi)$, como escoger una correspondiente secuencia de estados $Q = q_1 q_2 \dots q_T$, que sea óptima en el sentido de que mejor explique la observación generada.
- 3 **Aprendizaje-Entrenamiento** Como ajustar los parámetros del modelo $\lambda(A, B, \pi)$, para maximizar $P(O|\lambda)$.



Modelos Ocultos de Markov HMM

- 1 **Evaluación**, Dada una secuencia de observaciones $\vec{o}_1\vec{o}_2...\vec{o}_T$ y un HMM $\lambda(A, B, \pi)$, como calcular eficientemente $P(O|\lambda)$.
- 2 **Decodificación** Dada una secuencia de observaciones $\vec{o}_1\vec{o}_2...\vec{o}_T$ y un HMM $\lambda(A, B, \pi)$, como escoger una correspondiente secuencia de estados $Q = q_1q_2...q_T$, que sea óptima en el sentido de que mejor explique la observación generada.
- 3 **Aprendizaje-Entrenamiento** Como ajustar los parámetros del modelo $\lambda(A, B, \pi)$, para maximizar $P(O|\lambda)$.



Modelos Ocultos de Markov HMM

Topología de HMM para cada fonema

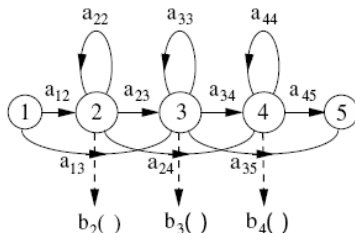


Figura: Topología de los HMM para cada fonema



Definición de la Estructura del HMM

```

~h "ah"
<BEGINHMM>
  <NUMSTATES> 5
  <STATE> 2 <NumMixes> 9
    <Mixture> 1 0.4
      <MEAN> 39
        -1.376990e+01 2.798773e+00 ... 5.959347e-04
      <VARIANCE> 39
        4.839668e+01 1.709005e+01 ... 7.078791e-02
    <Mixture> 1 0.01
      <MEAN> 39
        -1.376990e+01 2.798773e+00 ... 5.959347e-04
      <VARIANCE> 39
        4.839668e+01 1.709005e+01 ... 7.078791e-02
    .
    .
    .
  <STATE> 4 <NumMixes> 9
    <Mixture> 1 0.4
      <MEAN> 39
        -1.376990e+01 2.798773e+00 ... 5.959347e-04
      <VARIANCE> 39
        4.839668e+01 1.709005e+01 ... 7.078791e-02
    .
    .
    .
  <TRANSP> 5
    0.000000e+00 1.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
    0.000000e+00 6.000000e-01 4.000000e-01 0.000000e+00 0.000000e+00
    0.000000e+00 0.000000e+00 6.000000e-01 4.000000e-01 0.000000e+00
    0.000000e+00 0.000000e+00 0.000000e+00 7.000000e-01 3.000000e-01
    0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
  <ENDHMM>

```



Estimación de los Parámetros Iniciales de los HMM

ALGORITMO-ESTIMACIÓN-PARÁMETROS-INICIALES-HMM (HMM hmm , DATOS ENTRENAMIENTO E)

```

1  ▷ Calcular media y covarianza de los datos de entrenamiento
1   $\Sigma \leftarrow covarianza(E)$ 
2   $\mu \leftarrow media(E)$ 
3  ▷ Calcular numero de estados de todos los HMM's
3   $N \leftarrow length(hmm)$ 
4  for  $i \leftarrow 1$  to  $N$ 
5      do
6          ▷ Establecer la media y covarianza de todos los estados a la media y covarianza global
6           $hmm_i \leftarrow \mu$ 
7           $hmm_i \leftarrow \Sigma$ 

8  ▷ Segmentar los datos de manera uniforme
9   $segmento \leftarrow \frac{length(E)}{N}$ 
10 ▷ Asignar cada segmento a cada estado del HMM
11 for  $i \leftarrow 1$  to  $N$ 
12     do
13          $hmm_i \leftarrow [E_i \dots E_{i+segmento-1}]$ 

14 return  $hmm$ 

```



Entrenamiento de los HMM

ENTRENAMIENTO-HMM's(HMM h , DATOS ENTRENAMIENTO E , ARCHIVOS TRANSCRIPCIÓN)

```
1   $P \leftarrow \text{Read}(\text{archivoTranscripción})$ 
2  for  $e \leftarrow 0$  to  $E - 1$ 
3      do
4           $h[e] \leftarrow \text{Algoritmo Baum-Welch}(h[e], e, P_e)$ 
5  retornar a la línea 2 si fuera necesario
6  return  $h$ 
```



HMM para silencio y pausa

ALGORITMO-SILENCIO-PAUSA(HMM *hmm*)

- ▷ Extraer fonema SIL del HMM compuesto
- 1 *hmmSIL* \leftarrow *extraerHMMFonemaSil(hmm)*
- ▷ Agregar transiciones adicionales
- 2 *hmmSIL* \leftarrow *agregarTrans(estado2, estado4)*
- 3 *hmmSIL* \leftarrow *agregarTrans(estado4, estado2)*
- ▷ Crear HMM llamado sp
- 4 *hmmSP* \leftarrow *crearHMM()*
- 5 *hmmSP* \leftarrow *numeroEstados(3)*
- 6 *hmmSP* \leftarrow *agregarTrans(estado1, estado3)*
- 7 *hmmSP* \leftarrow *agregarTrans(estado2, estado2)*
- ▷ Compartir estado central del HMM sil con HMM sp
- 8 *hmmSILSP* \leftarrow *compartirEstado(hmmSIL₃hmmSP₂)*
- ▷ agregar fonema SILSP al HMM compuesto
- 9 *setHMMFonemaSilSp(hmm, hmmSILSP)*
- 10 **return** *hmm*



HMM para silencio y pausa

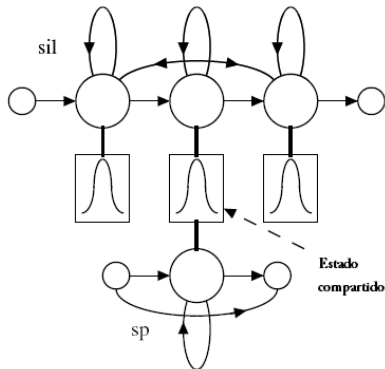


Figura: Los modelos de silencio **sil** y **sp**.



Descripción del Algoritmo

Descripción del Algoritmo

1 Preparación de los datos

- Diseño de la gramática
- Construcción del diccionario de pronunciación
- Obtención de los datos
- Construcción de los archivos de transcripción
- Extracción de características usando MFCC

2 Construcción de los HMM

- Definición de la estructura del HMM
- Estimación de los parámetros iniciales del HMM
- Entrenamiento de los HMM
- Establecer HMM para silencio y pausa

3 Creación de fonemas dependientes del contexto

- Creación de los trifenemas
- Clustering de estados

4 Decodificación

- Algoritmo Token-passing



Creación de los Trifonemas

-CREACION-TRIFONEMAS(HMM-COMPUESTO *hmm*, ARCHIVOTranscripción)

```

  ▷ Crear transcripciones de trifonemas a partir de las transcripciones de fonemas
1   $P \leftarrow \text{read}(\text{archivoTranscripcin})$ 
2   $N \leftarrow \text{length}(hmm)$ 
3  for  $i \leftarrow 1$  to  $N$ 
4    do ▷ Para todos los fonemas, crear trifonemas dependientes del contexto
      según el archivo de transcripción
5       $hmmTrifonema \leftarrow \text{crearTrifonema}(\text{fonema}_{\text{izquierdo}}(P_i), \text{fonema}_{\text{central}}(P_i), \text{fonema}_{\text{derecho}}(P_i))$ 
6       $hmmTrifonema \leftarrow \text{getMatrizTransicion}(\text{fonema}_{\text{central}}(P_i))$ 

7  ▷ Reestimar los parámetros del HMM del trifonema usando Baum Welch a partir de
  las transcripciones de los trifonemas
8  Baum-Welch(hmmTrifonema)
9  return hmmTrifonema

```



Creación de los Trifonemas

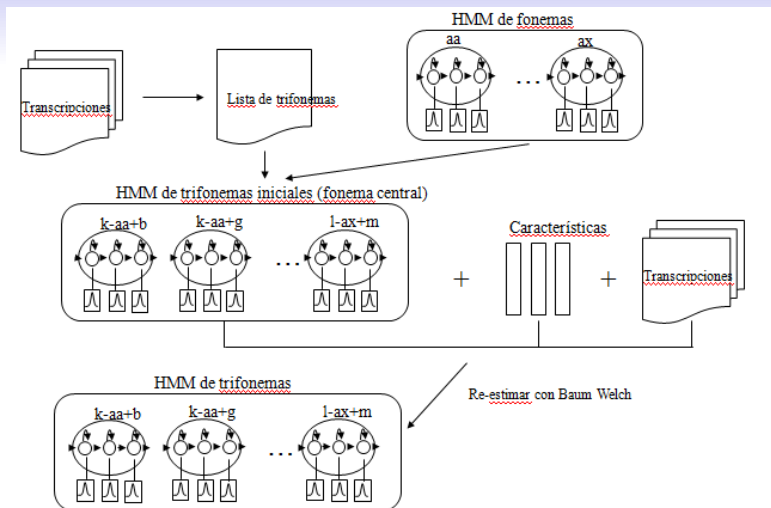


Figura: Construcción de los HMM de los trifonemas a partir de los HMM de los fonemas.



Creación de los Trifonemas

Lista de Trifonemas

"*/T0001.lab"

sil
t+eh
t-eh+l
eh-l+eh
l-eh+f
eh-f+oh
f-oh+n
oh-n+oh
n-oh
sp
dh+oh
dh-oh+s
oh-s
sp
s+eh
s-eh+y
eh-y+s
y-s
sp
k+w
k-w+ah
w-ah+t
ah-t+r
t-r+oh
r-oh
sp
t+r
t-r+eh
r-eh+s
eh-s
sp
uh+n
uh-n+oh
n-oh

dh+oh
dh-oh+s
oh-s
sp
th+ih
th-ih+ng
ih-ng+k
ng-k+oh
k-oh
sp
th+ih
th-ih+ng
ih-ng+k
ng-k+oh
k-oh
sp
k+w
k-w+ah
w-ah+t
ah-t+r
t-r+oh
r-oh
sp
uh+n
uh-n+oh
n-oh
sp
s+eh
s-eh+y
eh-y+s
y-s
sp
k+w
k-w+ah
w-ah+t
ah-t+r

eh-y+s
y-s
sp
th+eh
th-eh+r
eh-r+oh
r-oh
sp
th+ih
th-ih+ng
ih-ng+k
ng-k+oh
k-oh
sp
s+y
s-y+eh
y-eh+t
eh-t+eh
t-eh
sp
k+w
k-w+ah
w-ah+t
ah-t+r
t-r+oh
r-oh
sp
n+w
n-w+eh
w-eh+b
eh-b+eh
b-eh
sp
th+eh
th-eh+r
eh-r+oh



Clustering de estados

CLUSTERING-ESTADOS($N, hmmTrifonema$)

```
1  Crear un cluster por estado para todos los estados de hmmTrifonema
2   $n \leftarrow$  numero de clusters
3  while  $n > N$ 
4      do
5          Encontrar  $i$  y  $j$  para el cual  $g(i, j)$  es mínimo
6          Mezclar los clusters  $i$  y  $j$ 
7           $n \leftarrow n - 1$ 
8          Para cada cluster los estados deben compartir los mismos parámetros
9  return hmmTrifonema actualizado
```



Clustering de estados

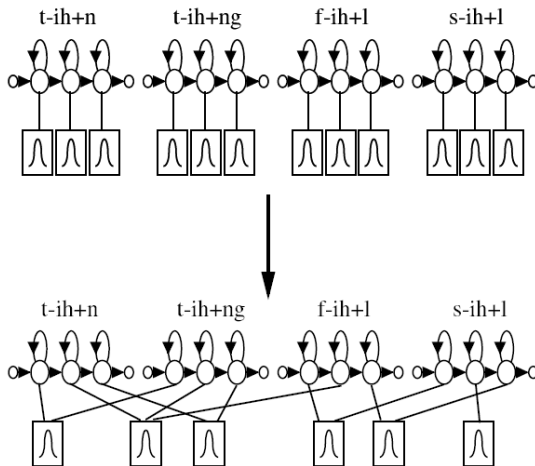


Figura: Clustering de los trifonemas.



Descripción del Algoritmo

Descripción del Algoritmo

1 Preparación de los datos

- Diseño de la gramática
- Construcción del diccionario de pronunciación
- Obtención de los datos
- Construcción de los archivos de transcripción
- Extracción de características usando MFCC

2 Construcción de los HMM

- Definición de la estructura del HMM
- Estimación de los parámetros iniciales del HMM
- Entrenamiento de los HMM
- Establecer HMM para silencio y pausa

3 Creación de fonemas dependientes del contexto

- Creación de los trifenemas
- Clustering de estados

4 Decodificación

- Algoritmo Token-passing



Decodificación

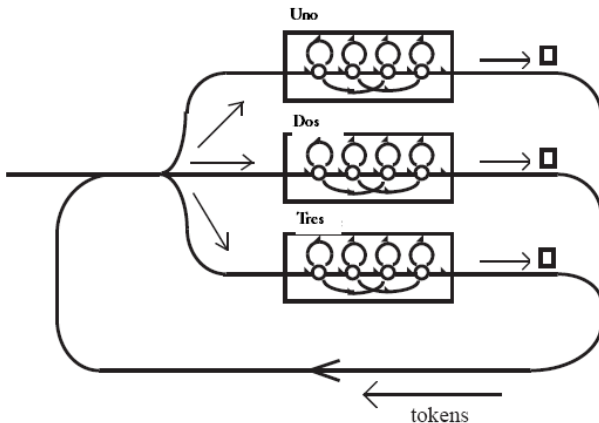


Figura: Esquema de decodificación usando el algoritmo token passing.



Algoritmo Propuesto

ALGORITMO-RECUPERACIÓN-INFORMACIÓN-TEXTOS-HABLADOS(G, D, N)

Entrada Gramática GLC G , Diccionario de pronunciación D ,
Número de Sentencias N , Periodo de muestreo T , tamaño del frame $\|frame\|$

Preparación de los datos

▷ Construcción del grafo de palabras

1 $Grafo \leftarrow$ ALGORITMO-PROCESAMIENTO-GRAMÁTICA (G)

▷ Construcción del diccionario de pronunciación

2 $Dic \leftarrow$ ALGORITMO-CONSTRUCCIÓN-DICCIONARIO-PRONUNCIACIÓN (D, G)

▷ Generación aleatoria y grabación de sentencias

3 **for** $i \leftarrow 1$ **to** N

4 **do**

5 $S_i \leftarrow$ ALGORITMO-GENERACIÓN-ALEATORIA-SENTENCIAS (G)

6 $xa_i \leftarrow$ GRABAR-SENTENCIA(S_i)

▷ Construcción de los archivos de transcripción

7 **for** $i \leftarrow 1$ **to** N

8 **do**

9 archivoTranscripción \leftarrow ALGORITMO-CONSTRUCCIÓN-ARCHIVOS-TRANSCRIPCIÓN

▷ Extracción de características usando MFCC

10 **for** $i \leftarrow 1$ **to** N

11 **do**

12 $mfcc_i \leftarrow$ ALGORITMO-MFCC($xa_i, T, \|frame\|$)



Diagrama de Flujo

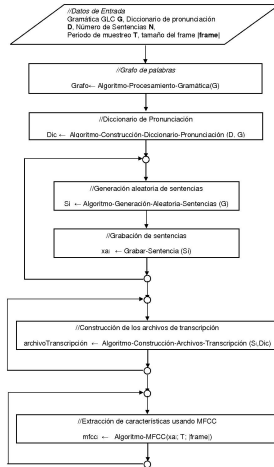


Figura: Diagrama de Flujo: Preparación de los Datos.

Algoritmo Propuesto

Construcción de los HMM's

▷ Definición de la estructura de los HMM's

```

13 for  $i \leftarrow 1$  to  $\text{length}(\text{extraerFonemas}(Dic))$ 
14     do ▷ Por cada fonema (incluido el fonema 'sil' para el silencio) del diccionario crear HMM
15          $hmm_i \leftarrow \text{crearHMM}(HMM\text{-Prototipo})$ 
16          $hmm_i \leftarrow \text{numeroEstados}(5)$  ▷ Definir un HMM con 5 estados
17         for  $e \leftarrow 2$  to 4
18             do ▷ Para los 3 estados intermedios
19                 ▷ Asignar un GMM con un arreglo de medias y un arreglo de matrices de covarianza
20                  $hmm_{i,e} \leftarrow \text{establecerGMM}(\mu[], \Sigma[])$ 

20          $hmm_i \leftarrow \text{matrizTransiciones}(A)$  ▷ Definir la matriz de transiciones del HMM

▷ Estimación de los parámetros iniciales de los HMM's
21 for  $i \leftarrow 2$  to  $\text{length}(\text{extraerFonemas}(Dic))$ 
22     do ▷ Creación de un solo HMM compuesto
23          $hmm_i \leftarrow \text{concatenar}(hmm_{i-1}, hmm_i)$ 

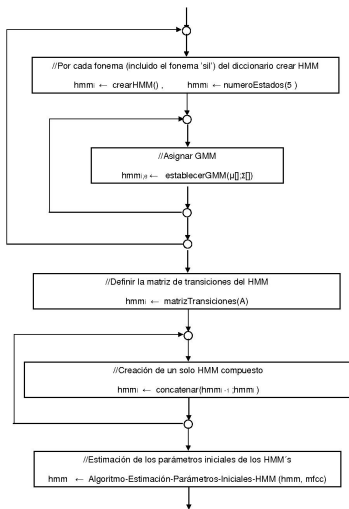
24  $hmm \leftarrow \text{ALGORITMO-ESTIMACIÓN-PARÁMETROS-INICIALES-HMM}(hmm, mfcc)$ 

▷ Entrenamiento de los HMM
25  $hmm \leftarrow \text{ALGORITMO-ENTRENAMIENTO-HMM's}(hmm, mfcc, \text{ARCHIVOTRANSCRIPCIÓN})$ 
▷ Establecer HMM para silencio y pausa
26  $hmm \leftarrow \text{ALGORITMO-SILENCIO-PAUSA}(hmm)$ 

```



Diagrama de Flujo



Algoritmo Propuesto

Creación de los fonemas dependientes del contexto

▷ Creación de los trifenemas

27 $hmmTri fonema \leftarrow CREACION-TRIFONEMAS(hmm, ARCHIVOTranscripción)$

▷ Clustering de estados

28 $hmmTri fonema \leftarrow CLUSTERING-ESTADOS(N, hmmTri fonema)$

Decodificación

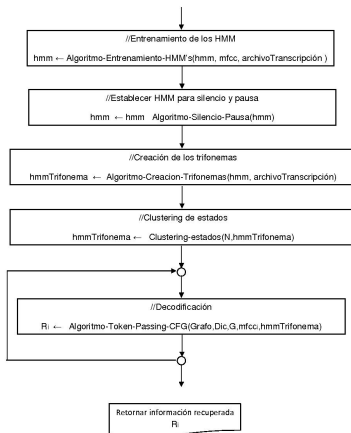
29 **for** $i \leftarrow 1$ **to** numeroArchivosRecuperar

30 $R_i \leftarrow ALGORITMO-TOKEN-PASSING-CFG(Grafo, Dic, G, mfcc_i, hmmTri fonema)$

31 **return** R



Diagrama de Flujo



Resultados

Correctas

$$\text{Correctas} = \frac{H}{N} \times 100 \quad (14)$$

Precisión

$$\text{Precisión} = \frac{H - I}{N} \times 100 \quad (15)$$



Resultados

Correctas

$$\text{Correctas} = \frac{H}{N} \times 100 \quad (14)$$

Precisión

$$\text{Precisión} = \frac{H - I}{N} \times 100 \quad (15)$$



Resultados por Sentencias

Sentencia	Correctas	Precisión	H	E	S	I	N
1	100.00	90.48	21	0	0	2	21
2	100.00	33.33	3	0	0	2	3
3	100.00	84.62	13	0	0	2	13
4	100.00	0.00	2	0	0	2	2
5	100.00	0.00	2	0	0	2	2
6	100.00	0.00	2	0	0	2	2
7	100.00	33.33	3	0	0	2	3
8	100.00	80.00	10	0	0	2	10
9	100.00	0.00	2	0	0	2	2
10	100.00	0.00	2	0	0	2	2
.
.
.
190	100.00	33.33	3	0	0	2	3
191	100.00	0.00	2	0	0	2	2
192	100.00	0.00	2	0	0	2	2
193	100.00	50.00	4	0	0	2	4
194	100.00	33.33	3	0	0	2	3
195	100.00	0.00	2	0	0	2	2
196	100.00	0.00	2	0	0	2	2
197	100.00	33.33	3	0	0	2	3
198	100.00	90.91	22	0	0	2	22
199	100.00	0.00	2	0	0	2	2
200	100.00	66.67	6	0	0	2	6

Cuadro: Resultados por sentencias, resultados de las primeras 50 sentencias.



Resultados Generales variando parámetros: Penalidad por inserción y factor de escala

p	q	Correctas	Precisión	H	E	S	I	N
0	5	99.83	64.96	1148	0	2	401	1150
0	10	99.83	64.96	1148	0	2	401	1500
0	20	99.83	64.96	1148	0	2	401	1150
0	40	99.83	64.96	1148	0	2	401	1150
0	75	99.83	64.96	1148	0	2	401	1150
0	100	99.83	64.96	1148	0	2	401	1150
5	0	99.83	64.96	1148	0	2	401	1150
10	0	99.83	64.96	1148	0	2	401	1150
20	0	99.83	64.96	1148	0	2	401	1150
40	0	99.83	64.96	1148	0	2	401	1150
75	0	99.83	64.43	1148	0	2	407	1150
100	0	99.83	63.65	1148	0	2	416	1150
5	95	99.83	64.96	1148	0	2	401	1150
20	80	99.83	64.96	1148	0	2	401	1150
30	70	99.83	64.96	1148	0	2	401	1150
45	55	99.83	64.96	1148	0	2	401	1150
50	50	99.83	64.87	1148	0	2	402	1150

Cuadro: Resultados para experimentos variando los valores: **penalidad por inserción de palabra** y el **factor escala de la gramática**.



Evaluación del Algoritmo usando Validación Cruzada k-fold

iter	1	2	3	4	5	6	7	8	9	10
EI	0.68	0.17	0.10	3.00	6.65	0.18	0.20	2.65	1.55	1.57
EP	0.68	0.43	0.32	0.99	2.12	1.80	1.57	1.70	1.69	1.68
iter	11	12	13	14	15	16	17	18	19	20
EI	0.17	0.17	0.17	0.13	0.15	1.00	4.53	0.16	0.17	0.17
EP	1.54	1.42	1.33	1.24	1.17	1.16	1.36	1.29	1.23	1.18

Cuadro: Error del clasificador obtenido usando validación cruzada k-fold



Evaluación del Algoritmo usando Validación Cruzada k-fold

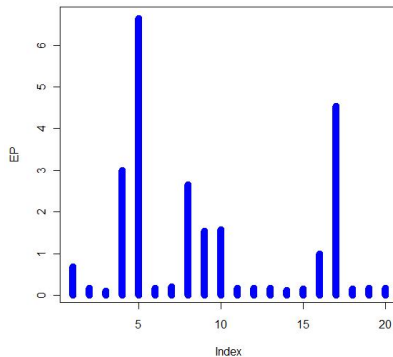


Figura: Error por iteración del k-fold.



Evaluación del Algoritmo usando Validación Cruzada k-fold

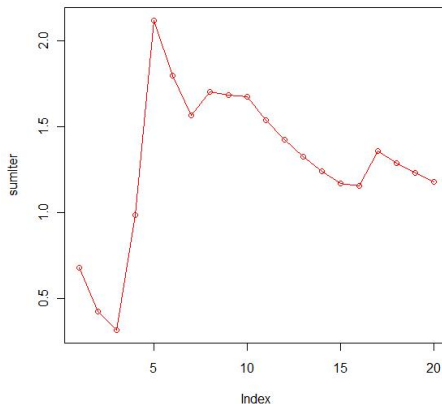


Figura: Error promedio por iteración del k-fold.



Prueba de Significancia Estadística de t-student

1 Nivel de significancia $\alpha = 0,05$

2 Hipótesis Nula.

$$H_0 : \mu \leq 0,05$$

(16)



Prueba de Significancia Estadística de t-student

1 Nivel de significancia $\alpha = 0,05$

2 Hipótesis Nula.

$$H_0 : \mu \leq 0,05 \quad (16)$$

3 Hipótesis Alternativa

$$H_1 : \mu > 0,05 \quad (17)$$



Prueba de Significancia Estadística de t-student

1 Nivel de significancia $\alpha = 0,05$

2 Hipótesis Nula.

$$H_0 : \mu \leq 0,05 \quad (16)$$

3 Hipótesis Alternativa

$$H_1 : \mu > 0,05 \quad (17)$$

4 Estadístico de Prueba (contraste unilateral)t:

$$\begin{aligned} t &= \frac{(\bar{x} - \mu)}{\frac{S}{\sqrt{n}}} \\ &= \frac{(0,0017 - 0,05)}{\frac{0,17}{\sqrt{200}}} \\ &= -0,0200901515 \end{aligned}$$



Prueba de Significancia Estadística de t-student

1 Nivel de significancia $\alpha = 0,05$

2 Hipótesis Nula.

$$H_0 : \mu \leq 0,05 \quad (16)$$

3 Hipótesis Alternativa

$$H_1 : \mu > 0,05 \quad (17)$$

4 Estadístico de Prueba (contraste unilateral)t:

$$\begin{aligned} t &= \frac{(\bar{x} - \mu)}{\frac{S}{\sqrt{n}}} \\ &= \frac{(0,0017 - 0,05)}{\frac{0,17}{\sqrt{200}}} \\ &= -0,0200901515 \end{aligned}$$

5 S varianza del error de las muestras, n es el número de muestras.



Prueba de Significancia Estadística de t-student

1 Nivel de significancia $\alpha = 0,05$

2 Hipótesis Nula.

$$H_0 : \mu \leq 0,05 \quad (16)$$

3 Hipótesis Alternativa

$$H_1 : \mu > 0,05 \quad (17)$$

4 Estadístico de Prueba (contraste unilateral)t:

$$\begin{aligned} t &= \frac{(\bar{x} - \mu)}{\frac{S}{\sqrt{n}}} \\ &= \frac{(0,0017 - 0,05)}{\frac{0,17}{\sqrt{200}}} \\ &= -0,0200901515 \end{aligned}$$

5 S varianza del error de las muestras, n es el número de muestras.



Prueba de Significancia Estadística de t-student

1 Región de Aceptación

$$(-\infty, t_\alpha)$$

$$(-\infty, 0,254)$$

(18)

- 2
- Aceptar H_0 si $t \in (-\infty, 0,254)$
- Rechazar H_0 si $t \notin (-\infty, 0,254)$



Prueba de Significancia Estadística de t-student

1 Región de Aceptación

$$\begin{aligned} &(-\infty, t_\alpha) \\ &(-\infty, 0,254) \end{aligned} \tag{18}$$

- 2
 - Aceptar H_0 si $t \in (-\infty, 0,254)$
 - Rechazar H_0 si $t \notin (-\infty, 0,254)$
- 3 Finalmente como $t = -0,0200901515$ se encuentra en la región de aceptación tenemos que se acepta la hipótesis nula H_0



Prueba de Significancia Estadística de t-student

1 Región de Aceptación

$$\begin{aligned} &(-\infty, t_{\alpha}) \\ &(-\infty, 0,254) \end{aligned} \tag{18}$$

- 2
 - Aceptar H_0 si $t \in (-\infty, 0,254)$
 - Rechazar H_0 si $t \notin (-\infty, 0,254)$
- 3 Finalmente como $t = -0,0200901515$ se encuentra en la región de aceptación tenemos que se acepta la hipótesis nula H_0



Discusión

sentencia 126						
testT0126.lab	TELEFONO CUATRO TRES CUATRO DOS NUEVE UNO OCHO DOS DOS					
testT0126.rec	SENT-START TELEFONO CUATRO TRES CUATRO DOS NUEVE UNO OCHO DOS DO					
Correctas	Precisión	H	E	S	I	M
100	80	10	0	0	2	1

sentencia 29						
testT0029.lab	LLAMAR LUIS					
testT0029.rec	SENT-START MARCAR LUIS SENT-END					
Correctas	Precisión	H	E	S	I	M
50.00	-50.00	1	0	1	2	2

sentencia 149						
testT0149.lab	TELEFONO SIETE CINCO					
testT0149.rec	SENT-START TELEFONO CERO CINCO SENT-END					
Correctas	Precisión	H	E	S	I	M
66.67	0.00	2	0	1	2	3



Discusión

- La Tabla 2 muestra 17 experimentos realizados variando los parámetros p y q . Cada experimento fue hecho usando 200 sentencias generadas aleatoriamente, con un promedio de 10 palabras por sentencia, esto hace un total de 17×200 sentencias y $17 \times 200 \times 10$ palabras en promedio. Analizando la precisión podemos notar que los valores de p muy altos cercanos a 100 decrementan la precisión. El valor de Correctas fue 99.83 %, el valor Precisión está en un rango de 63.65 a 64.96.
- La Tabla 3 describe la validación del clasificador. Estos resultados muestran que el error promedio del es de 1.18 % después de 20 iteraciones. Esta validación experimental brinda una idea del desempeño del clasificador usando los datos originales como test y entrenamiento escogidos de manera aleatoria.



Discusión

- La Tabla 2 muestra 17 experimentos realizados variando los parámetros p y q . Cada experimento fue hecho usando 200 sentencias generadas aleatoriamente, con un promedio de 10 palabras por sentencia, esto hace un total de 17×200 sentencias y $17 \times 200 \times 10$ palabras en promedio. Analizando la precisión podemos notar que los valores de p muy altos cercanos a 100 decrementan la precisión. El valor de Correctas fue 99.83 %, el valor Precisión está en un rango de 63.65 a 64.96.
- La Tabla 3 describe la validación del clasificador. Estos resultados muestran que el error promedio del es de 1.18 % después de 20 iteraciones. Esta validación experimental brinda una idea del desempeño del clasificador usando los datos originales como test y entrenamiento escogidos de manera aleatoria.
- La prueba de significancia de t-student proporciona una estimación de la frecuencia con que podrían ocurrir por azar los resultados experimentales. Los resultados de una prueba de este tipo se plantean como una prueba de probabilidad, indicando las posibilidades de que la diferencia observada se haya debido al azar. Fue aceptada la hipótesis nula con un nivel de significancia de 0.05.



Discusión

- La Tabla 2 muestra 17 experimentos realizados variando los parámetros p y q . Cada experimento fue hecho usando 200 sentencias generadas aleatoriamente, con un promedio de 10 palabras por sentencia, esto hace un total de 17×200 sentencias y $17 \times 200 \times 10$ palabras en promedio. Analizando la precisión podemos notar que los valores de p muy altos cercanos a 100 decrementan la precisión. El valor de Correctas fue 99.83 %, el valor Precisión está en un rango de 63.65 a 64.96.
- La Tabla 3 describe la validación del clasificador. Estos resultados muestran que el error promedio del es de 1.18 % después de 20 iteraciones. Esta validación experimental brinda una idea del desempeño del clasificador usando los datos originales como test y entrenamiento escogidos de manera aleatoria.
- La prueba de significancia de t-student proporciona una estimación de la frecuencia con que podrían ocurrir por azar los resultados experimentales. Los resultados de una prueba de este tipo se plantean como una prueba de probabilidad, indicando las posibilidades de que la diferencia observada se haya debido al azar. Fue aceptada la hipótesis nula con un nivel de significancia de 0.05.
- El tiempo de ejecución del algoritmo es lineal en la etapa de decodificación, es dominado en la etapa de entrenamiento por el entrenamiento de los HMM.



Discusión

- La Tabla 2 muestra 17 experimentos realizados variando los parámetros p y q . Cada experimento fue hecho usando 200 sentencias generadas aleatoriamente, con un promedio de 10 palabras por sentencia, esto hace un total de 17×200 sentencias y $17 \times 200 \times 10$ palabras en promedio. Analizando la precisión podemos notar que los valores de p muy altos cercanos a 100 decrementan la precisión. El valor de Correctas fue 99.83 %, el valor Precisión está en un rango de 63.65 a 64.96.
- La Tabla 3 describe la validación del clasificador. Estos resultados muestran que el error promedio del es de 1.18 % después de 20 iteraciones. Esta validación experimental brinda una idea del desempeño del clasificador usando los datos originales como test y entrenamiento escogidos de manera aleatoria.
- La prueba de significancia de t-student proporciona una estimación de la frecuencia con que podrían ocurrir por azar los resultados experimentales. Los resultados de una prueba de este tipo se plantean como una prueba de probabilidad, indicando las posibilidades de que la diferencia observada se haya debido al azar. Fue aceptada la hipótesis nula con un nivel de significancia de 0.05.
- El tiempo de ejecución del algoritmo es lineal en la etapa de decodificación, y es dominado en la etapa de entrenamiento por el entrenamiento de los HMM.



Conclusiones

- El error del clasificador muy pequeño, 0.0 % en los experimentos de sentencia por palabras, 0.17 % variando los parámetros de penalidad por inserción y factor de escala y 1.18 % usando validación cruzada k-fold.
- Importancia del modelo del lenguaje, el valor Precisión muestra esta importancia.



Conclusiones

- El error del clasificador muy pequeño, 0.0 % en los experimentos de sentencia por palabras, 0.17 % variando los parámetros de penalidad por inserción y factor de escala y 1.18 % usando validación cruzada k-fold.
- Importancia del modelo del lenguaje, el valor Precisión muestra esta importancia.
- El análisis del tiempo de ejecución del algoritmo indica que el proceso de recuperación es rápido, siendo del orden $\theta(tk)$.



Conclusiones

- El error del clasificador muy pequeño, 0.0 % en los experimentos de sentencia por palabras, 0.17 % variando los parámetros de penalidad por inserción y factor de escala y 1.18 % usando validación cruzada k-fold.
- Importancia del modelo del lenguaje, el valor Precisión muestra esta importancia.
- El análisis del tiempo de ejecución del algoritmo indica que el proceso de recuperación es rápido, siendo del orden $\theta(tk)$.
- Modelos de recuperación simples como archivos de grabaciones telefónicas son relativamente menos complicados que grabaciones de conversaciones reales con ruido de fondo incluido, como consecuencia, es posible aplicar esta tecnología a situaciones donde se tenga un número limitado de términos y el entorno sea limpio de ruido.



Conclusiones

- El error del clasificador muy pequeño, 0.0 % en los experimentos de sentencia por palabras, 0.17 % variando los parámetros de penalidad por inserción y factor de escala y 1.18 % usando validación cruzada k-fold.
- Importancia del modelo del lenguaje, el valor Precisión muestra esta importancia.
- El análisis del tiempo de ejecución del algoritmo indica que el proceso de recuperación es rápido, siendo del orden $\theta(tk)$.
- Modelos de recuperación simples como archivos de grabaciones telefónicas son relativamente menos complicados que grabaciones de conversaciones reales con ruido de fondo incluido, como consecuencia, es posible aplicar esta tecnología a situaciones donde se tenga un número limitado de términos y el entorno sea limpio de ruido.



Recomendaciones

- Modelos del lenguaje más sofisticados como los n-grams.
- Ambientes con ruido, varias voces al mismo tiempo.



Recomendaciones

- Modelos del lenguaje más sofisticados como los n-grams.
- Ambientes con ruido, varias voces al mismo tiempo.
- Modelos cuando no se tiene información apriori del lenguaje.



Recomendaciones

- Modelos del lenguaje más sofisticados como los n-grams.
- Ambientes con ruido, varias voces al mismo tiempo.
- Modelos cuando no se tiene información apriori del lenguaje.
- Extender a sistemas independientes del hablante.



Recomendaciones

- Modelos del lenguaje más sofisticados como los n-grams.
- Ambientes con ruido, varias voces al mismo tiempo.
- Modelos cuando no se tiene información apriori del lenguaje.
- Extender a sistemas independientes del hablante.



Referencias

**Steven B. Davis and Paul Mermelstein.**

Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences.

pages 65–74, 1990.

**A. A. Markov.**

An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology.

Math. Stat., 37(v37.):360–363., 1967.

**Paul Purdom.**

A sentence generator for testing parsers.

BIT Numerical Mathematics, 12:366–375, 1972.

10.1007/BF01932308.

**Lawrence R. Rabiner.**

A tutorial on hidden markov models and selected applications in speech recognition.

pages 267–296, 1990.

