2. El Problema de Reconocimiento de Voz

Para poder hablar de reconocimiento de voz es necesario entender algunos de los conceptos fundamentales que componen a un reconocedor de voz. Un reconocedor de voz es un dispositivo que transcribe voz a texto [1]. Particularmente existe una señal acústica a la cual se le extraen las partes más importantes o características mediante procesamiento de señales. Una vez preparados los datos para una búsqueda de patrones o correspondencia de patrones, el reconocedor de voz toma un conjunto de vectores de características como entrada y usa tres tipos de conocimiento para determinar la cadena de palabras. Estos son: el modelo acústico, el modelo del lenguaje y el léxico.

El proceso de búsqueda de patrones toma la secuencia de características, el modelo del lenguaje, el modelo acústico, y el léxico y entonces determina la cadena *más probable* de palabras dadas estas fuentes de conocimiento.

En este capítulo se definirán algunos de los fundamentos o partes más importantes para un reconocedor así como la formulación probabilística del problema de reconocimiento de voz y detalles de cada uno de los componentes de un reconocedor de voz.

2.1 Fonética y Fonología

Una parte importante de los sistemas de reconocimiento de voz y texto-a-voz (text-to-speech): es como las palabras son pronunciadas en términos de unidades individuales de

voz llamadas fonemas. Un sistema de reconocimiento de voz necesita tener la pronunciación para cada una de las palabras que pueda reconocer. Como veremos un fono [*t*] es pronunciado muy diferente en distintos ambientes fonéticos. *Fonología* es el área de la lingüística que describe la forma sistemática en que los sonidos son diferentemente realizados en distintos ambientes, y cómo este sistema de sonidos está relacionado con el resto de la gramática.

El estudio de la pronunciación de las palabras es parte del campo de la *fonética*, el estudio de los sonidos de la voz usada en los idiomas del mundo. Modelamos la pronunciación de una palabra como una cadena de símbolos los cuales representan fonos o segmentos. Un fono es un sonido de voz; representamos fonos con símbolos fonéticos que mantengan algún parecido a una letra en un idioma alfabético como el Español. Por ejemplo, hay un fono representado por *m* que de hecho corresponde a la letra *m* y un fono representado por *t* que normalmente corresponde a la letra *t*. Como veremos después, los fonos tienen muchas más variantes que las letras [8].

Otra parte que debemos tomar en cuenta es la *fonética articulatoria*, que es el estudio de cómo los fonos son producidos. Este tema lo discutiremos en el capítulo 4 donde definiremos cada uno de los fonos a utilizar para nuestro sistema de reconocimiento de voz en español.

La descripción anterior es necesaria para poder mostrar como es que un reconocedor trata con unidades básicas llamadas fonemas y que están dadas por la descomposición de la

señal de voz en vectores de características que son una de las partes más importantes en el proceso de reconocimiento de voz.

A continuación se dará la formulación matemática para el diseño de un reconocedor de voz.

2.2 Fundamentos Matemáticos para el Reconocimiento de Voz

2.2.1 Teorema de Bayes[1]

Sea A la evidencia acústica (datos) la base sobre de la cual el reconocedor hará su decisión acerca de qué palabras fueron habladas. Al tratar con las computadoras, la señal es digitalizada, entonces sin pérdida de generalidad asumimos que A es una secuencia de símbolos tomados de algún alfabeto A.

$$A = a_1, a_2, ..., a_m \qquad a_i \in A \tag{2.1}$$

Los símbolos a_i pueden ser generados en el tiempo, como está indicado en el índice i.

Sea

$$W = w_1, w_2, ..., w_n \qquad W_i \in V \tag{2.2}$$

denota una cadena de n palabras, cada una perteneciendo a un vocabulario fijo y conocido V.

[1] El material en esta sección y dos de las siguientes está basado en el primer capítulo de [1]

Si $P(\mathrm{W}|\mathrm{A})$ denota la probabilidad de que las palabras W fueron habladas, dada la observación de la evidencia A, entonces el reconocedor debiera decidir a favor de una cadena $\hat{\mathrm{W}}$ satisfaciendo

$$\hat{\mathrm{W}} = \arg\max_{w} P(\mathrm{W}|\mathrm{A}) \tag{2.3}$$

Esto es, el reconocedor escogerá la cadena más probable dada la evidencia acústica observada.

Subrayando la fórmula (2.3) suponemos que todas las palabras de un mensaje son igualmente importantes para el usuario, esto es, que el no-reconocimiento no acarrea una penalización diferente dependiendo sobre que palabra no fue reconocida.

Esta fórmula de Bayes de la teoría de probabilidad nos permite rescribir la probabilidad del lado derecho de (2.3) como

$$P(W \mid A) = \frac{P(W)P(A \mid W)}{P(A)} \tag{2.4}$$

donde $P(\mathrm{W})$ es la probabilidad de que la cadena W será pronunciada, $P(\mathrm{A}|\mathrm{W})$ es la probabilidad de que cuando el locutor diga W se observe la evidencia acústica A, y $P(\mathrm{A})$ es la probabilidad promedio de que A será observada. Esto es,

$$P(A) = \sum_{W'} P(W')P(A \mid W') \tag{2.5}$$

Ya que la maximización en (2.3) es llevada a cabo con la variable fija A, entonces de (2.3) y (2.4) el objetivo del reconocedor es encontrar la cadena $\hat{W}$ que maximice el producto de $P(W)P(A|W)$, esto es, satisface

$$\hat{W} = \arg\max_{w} P(W)\, P(A|W) \tag{2.6}$$

La formula (2.6) determina que procesos y componentes toman parte en el diseño de un reconocedor de voz. Posteriormente en este capítulo se definirán cada uno de estos componentes.

2.2.2 Cadenas de Markov

Para poder definir un modelo oculto de Markov (HMM) es necesario definir una cadena de Markov.

Una cadena de Markov es un concepto bien estudiado y muy importante de la teoría de probabilidad. Se trata de una clase de procesos aleatorios (o estocásticos) que incorporan una mínima cantidad de memoria sin quedarse sin la misma. Aquí trataremos con variables aleatorias discretas y cadenas de Markov finitas.

Sea $X_1, X_2, ..., X_n, ...$ una secuencia de variables aleatorias tomando sus valores en el mismo alfabeto finito $\mathcal{H}=\{1,2,...,c\}$. Si nada más es dicho, entonces la fórmula de Bayes aplica:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_1 \mid X_1, X_2, ..., X_{i-1}) \tag{2.7}$$

Las variables aleatorias forman una cadena de Markov, sin embargo, sí

$$P(X_1, X_2, ..., X_{i-1}) = P(X_i \mid X_{i-1}) \text{ para todos los valores de } i \tag{2.8}$$

Como una consecuencia, para las cadenas de Markov

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_1 \mid X_{i-1}) \tag{2.9}$$

Además tales procesos aleatorios tienen la memoria más simple: El valor en el tiempo *i* depende solamente del valor en el tiempo precedente y nada de lo que fue anteriormente.

La cadena de Markov es invariante en el tiempo u homogénea si a pesar del valor de tiempo con índice *i*,

$$P(X_i = x' \mid X_{i-1} = x) = p(x' \mid x) \qquad \text{para todo x, x'} \in \mathcal{H} \tag{2.10}$$

p(x'|x) es llamada la *función de transición* y puede ser presentada como una matriz *c x c*. Por supuesto, p(x'|x), la cual especifica todo acerca del proceso aleatorio, debe satisfacer para todo $x \in \mathcal{H}$ las condiciones usuales

$$\sum_{x \in H} p(x' | x) = 1 \qquad p(\text{x'}|\text{x}) \geq 0 \qquad \text{x'} \in \mathcal{H}$$

Se pueden ver los valores de X_i como estados y además en la cadena de Markov como un *proceso de estado finito* con transiciones entre estados especificados por la función p(x'|x). Si el alfabeto $\mathcal{H}$ no es muy grande, entonces la cadena puede ser completamente especificada intuitivamente por un diagrama de instancias como en la figura 2.1, el cual aplica a un espacio de estado de tamaño c=3. En la figura, las flechas con transiciones de valores de probabilidad marcan las transiciones entre estados. Algunas de la transiciones no se muestran, implicando que $p(1|2)=p(2|2)=p(3|3)=0$. La forma de la figura nos muestra por qué se pudiera llamar al proceso una *cadena*.

La restricción aparente de las cadenas de Markov de memoria en un paso es engañosa. En principio, las cadenas de Markov son capaces de modelar procesos de complejidad arbitraria. En realidad, consideremos un proceso $Z_1, Z_2, ..., Z_n, ...$ de longitud de memoria k, esto es,

$$P(Z_1, Z_2, ..., Z_n, ...) = \prod_i P(Z_i \mid Z_{i-k}, Z_{i-k+1}, ..., Z_{i-1})$$

Si definimos nuevas variables aleatorias

$$X_i = Z_{i-k+1}, Z_{i-k+2}, ..., Z_i \tag{2.11}$$

entonces la secuencia-*Z* especifica la secuencia-*X*, y vice versa, y además el proceso *X* es una cadena de Markov para cuya fórmula (2.9) se mantiene. Por supuesto, el espacio de estados $\mathcal{H}$ resultante es muy grande, y el proceso *Z* puede ser caracterizado directamente en un forma mucho más simple que vía transformación (2.11).

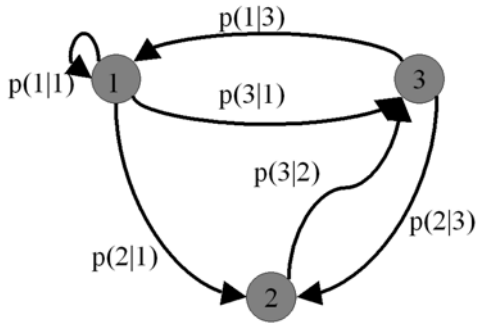

Figura 2.1 Cadena de Markov de tres estados

2.2.3 Modelos Ocultos de Markov

Se podría permitir más libertad para el proceso aleatorio mientras evitamos una complicación sustancial a la estructura básica de las cadenas de Markov. Podemos ganar esta libertad dejando generar a los estados de la cadena datos observables mientras escondemos la secuencia de estados asimismo del observador.

Además definiremos

1. Un alfabeto de salida $\mathcal{Y}$= {0,1,...,*b*-1}
2. Un espacio de estados $\mathcal{S}$ = {1,2,...,*c*} con un único estado inicial S_o.
3. Una distribución de probabilidad de transiciones entre estados *p*(s'|s), y

4. Una distribución de probabilidad de salida q(y|s,s') asociada con transiciones del estado s al estado s'.

Entonces la probabilidad de observar una cadena de salida HMM $y_1, y_2, ..., y_k$ está dada por

$$p(y_1, y_2, ..., y_k) = \sum_{s_1,...,s_i} \prod_{i=1}^{k} p(s_i \mid s_{i-1}) q(y_i \mid s_{i-1}, s_i) \tag{2.12}$$

La figura 2.2 es un ejemplo de un HMM con b=2 y c=3. Ahí, hemos añadido las distribuciones de salida q(y|s,s') a las transiciones y hemos omitido las probabilidades de transición p(s'|s).

Aunque el proceso de estado tiene solamente un-paso de memoria,

$$p(s_1, s_2, ..., s_k) = \prod_{i=1}^{k} p(s_i \mid s_{i-1})$$

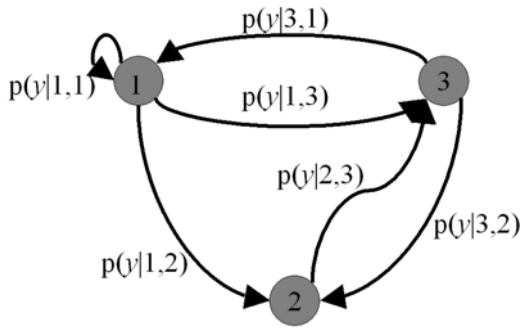

Figura 2.2 Modelo oculto de Markov de tres estados con salidas $y \in \{0,1\}$

la memoria de observables es ilimitada (excepto en casos degenerados). Esto es, en general, para todo $k \geq 2$,

$$P(y_{k+1} \mid y_1, y_2, ..., y_k) \neq P(y_{k+1} \mid y_j, ..., y_k) \qquad k \geq j \geq 2$$

Frecuentemente consideraremos conveniente al HMM como tener múltiples transiciones entre pares de estados, cada uno asociado con un diferente símbolo generado de salida, con probabilidad 1, cuando esa transición sea tomada. La figura 2.3 da un ejemplo de que se puede generar el mismo proceso aleatorio como en la figura 2.2.[2]

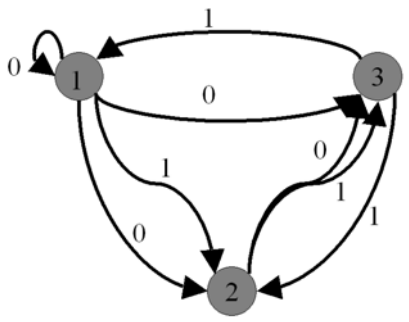

Figura 2.3 Representación de un modelo oculto de Markov ligando producciones a las transiciones

Esta vista tiene la ventaja de permitirnos proveer cada transición del HMM entero con un identificador diferente t y definir una función de salida $Y(t)$ que asigna a t un único símbolo de salida tomado del alfabeto $\mathcal{Y}$.

Entonces denotamos con $L(t)$ y $R(t)$ los estados origen y destino de la transición t, respectivamente. Denotamos $p(t)$ la probabilidad de que el estado $L(t)$ sea excitado vía la transición t, así que para todo $s \in \mathcal{S}$,

[2] Asumiendo que en la figura 2.2 $q(1|1,1)=q(1|1,3)=q(0|3,1)=q(0|3,2)=0$

$$\sum_{t:L(t)=s} p(t) = 1$$

La correspondencia entre las dos formas de ver un HMM está dada por la relación

$$p(t) = q(Y(t)|L(t), R(t))p(R(t)|L(t)) \tag{2.13}$$

Cuando las transiciones determinan las salidas, la probabilidad $P(y_1, y_2, ..., y_k)$ llega a ser igual a la suma de los productos $\Pi_{i=1}^{k} p(t_i)$ sobre todas las secuencias de transiciones $t_1, ..., t_k$ tal que $L(t_1) = s_0, Y(t_i) = y_i$, y $R(t_i) = L(t_{i+1})$ para i=1,...,k, o, formalmente,

$$P(y_1, y_2, ..., y_k) = \sum_{s(y_1, y_2, ..., y_k)} \prod_{i=1}^{k} p(t_i)$$

donde

$$S(y_1, y_2, ..., y_k) = \{t_1, ..., t_k : L(t_1) = s_0, Y(t_i) = y_i, R(t_i) = L(t_{i+1})$$
$$\text{para } i = 1, \ldots, k\}$$

2.2.4 Los Tres Problemas Básicos para HMMs

Dada la forma de los modelos ocultos de Markov de la sección anterior, hay tres problemas básicos de interés que deben ser resueltos para que el modelo sea útil en aplicaciones del mundo real [1,2,3]. Los problemas son los siguientes:

1. Problema de *evaluación*, dado un modelo y una secuencia de observaciones, ¿Cómo podemos calcular la probabilidad de que una secuencia observada fue producida por el modelo? También lo podemos ver como un problema de resultados, que tan bien, dado un modelo concuerda con una secuencia de observaciones. El último punto de vista es muy útil. Por ejemplo, si consideramos el caso en el cual tratamos de seleccionar entre varios modelos, la solución al problema 1 nos permite escoger el modelo que mejor concuerde con las observaciones.
2. En este problema tratamos de revelar o destapar la parte oculta del modelo, i.e., encontrar la secuencia *correcta* de estados. Debería ser claro para todos pero en el caso de modelos degenerados, no existe una secuencia de estado correcta para ser encontrada. Por lo tanto para situaciones prácticas, normalmente usamos un criterio de optimización para resolver este problema lo mejor que sea posible.
3. Para el último problema tratamos de optimizar los parámetros del modelo para que describa mejor como ocurre una secuencia de observaciones dada. La secuencia de observaciones usada para ajustar los parámetros del modelo es llamada una secuencia de entrenamiento ya que es usada para entrenar el HMM. El problema de entrenamiento es crucial para muchas aplicaciones de HMM, ya que nos permite adaptar los parámetros del modelo de forma óptima para observar los datos de entrenamiento, i.e., crear mejores modelos para fenómenos reales.

Para cada uno de los problemas existen soluciones que no se verán en este trabajo pero de manera sencilla podemos decir que para el problema evaluación de un HMM existe el algoritmo Forward el cual es $O(N^2T)$, donde T es la longitud de toda la secuencia posible de estados. Para el segundo problema existe el algoritmo de Viterbi que es del mismo orden

que el anterior y descodifica un HMM. En el problema de cómo estimar los parámetros de un HMM se usa el algoritmo de Baum-Welch o algoritmo forward-backward. La teoría de la solución de estos problemas no esta cubierta en este documento pero se pueden consultar en [2,3].

2.3 Componentes de un Reconocedor de Voz

La arquitectura del proceso de reconocimiento de voz está dada por cuatro elementos básicos [1,3,8,9]. Como hemos visto en la sección 2.2 de este capítulo, estos elementos se derivan de la fórmula (2.6). En la figura 2.4 podemos ver la arquitectura donde la función de cada uno de esos elementos se explica en los siguientes puntos.

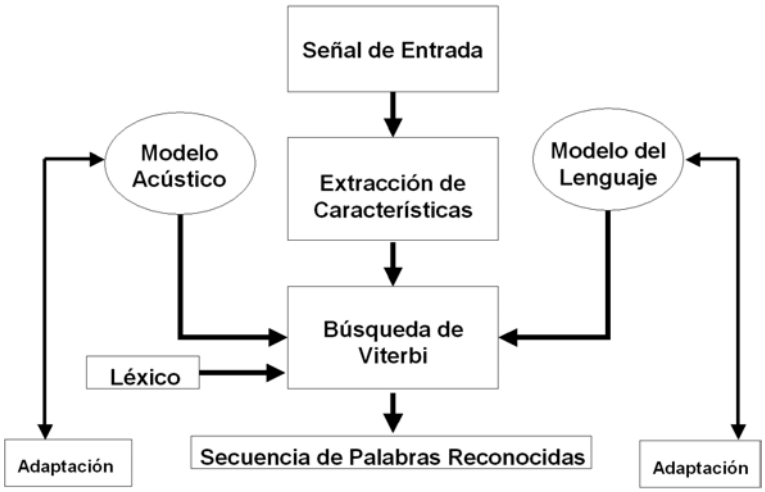

Figura 2.4 Arquitectura de un reconocedor de voz

En el proceso de reconocimiento de voz, primero es necesario decidir qué datos acústicos A serán observados. Esto es, necesitamos decidir en el inicio del sistema lo que transformará la presión de onda o sonido en símbolos a_i con los que tratará el reconocedor. Así que en principio todo este proceso incluye un micrófono cuya salida es una señal

eléctrica, las muestras de ejemplos de esa señal, y una manera de procesar el resultado de la secuencia de ejemplos. En este documento no se discutirá a fondo sobre procesamiento de señales ya que es un tema muy amplio y requeriría un mejor espacio para su explotación. Pero básicamente podemos darnos una idea de un procesador de señales como un dispositivo que genera vectores de valores σ_i en intervalos regulares de tiempo (e.g., unos cientos de veces por segundo). Los componentes de σ_i podrían ser valores de muestras de salidas de filtros de paso de banda aplicados a la señal que proviene del micrófono [1].

El prototipo de almacenamiento contiene un conjunto de vectores prototipo $\mathcal{R}=\{p_1, p_2, ..., p_k\}$ de la misma clase que σ_i. El *comparador* encuentra el elemento más cercano de $\mathcal{R}$ a σ_i, y el índice de ese elemento es el símbolo acústico a_i. Para ser precisos,

$$\hat{\boldsymbol{J}} = \arg\min_{j=1}^{k} d(\sigma_i, p_j) \tag{2.14}$$

y

$$a_i = \hat{\boldsymbol{J}} \tag{2.15}$$

En (14), $d(\ ,\ )$ denota un función de distancia.

Un método simple, llamado *Cuantización de Vectores (vector quantization*) [10], puede derivar el conjunto prototipo $\mathcal{R}=\{p_1, p_2, ..., p_k\}$ de los datos de voz.

2.3.1 Modelado Acústico

Teniendo en mente la fórmula (2.6), el reconocedor necesita ser capaz de determinar el valor de $P(\mathrm{A}|\mathrm{W})$ de la probabilidad de que cuando el locutor pronuncie la secuencia de palabras W el procesador acústico produzca los datos A.

Además para calcular *P*(A|W) necesitamos un modelo acústico estadístico de la interacción de los locutores con el procesador acústico. El total de fenómenos acústicos que estamos modelando involucra la forma en como el locutor pronuncia las palabras de W, el ambiente (ruido en el salón, eco, entre otros.), la colocación del micrófono y características, y el procesamiento acústico.

El modelo acústico normalmente empleado en los reconocedores de voz, se basa en los modelos ocultos de Markov, los cuales ya hemos discutido en las secciones anteriores de este capítulo. No hay que descartar otros modelos, por ejemplo aquéllos basados en redes neuronales artificiales o la deformación dinámica en el tiempo (*dynamic time warping*), ya que se pueden obtener otros resultados aún no estudiados [1,3].

2.3.2 Modelado del Lenguaje

La fórmula (2.6) además asume que es posible calcular para cada cadena W la probabilidad a priori[3] *P*(W) de que el locutor desee pronunciar W. La fórmula de Bayes nos permite varias descomposiciones de *P*(W). Dado que el reconocedor "naturalmente" desea expresar el texto en la secuencia en la que este fue hablado, usaremos la descomposición:

$$P(W) = \prod_{i=1}^{n} P(w_i \mid w_1,...,w_{i-1}) \tag{2.16}$$

El reconocedor además debe ser capaz de determinar los estimados de las probabilidades $P(w_i \mid w_1,...,w_{i-1})$. Usamos el término estimar a propósito, porque hasta para

[3] La probabilidad de un evento antes de que consideremos nuestro conocimiento

valores moderados de *i* y vocabularios de tamaño razonable, la probabilidad $P(w_i \mid w_1,...,w_{i-1})$ tiene muchos argumentos. De hecho, si $|\mathcal{V}|$ denota el tamaño del vocabulario, entonces para $|\mathcal{V}|$=20,000 e *i*=3, el número de argumentos es $8 \; x \; 10^{12}$.

Esto es, por supuesto, absurdo pensar que la elección del locutor de su *i-ésima* palabra depende de la *historia* entera $w_1,...,w_{i-1}$ de todo su discurso previo. Es además natural que para propósitos de la elección de w_i, la historia sea puesta en clases de equivalencia $\Phi(w_1,...,w_{i-1})$. Además en realidad la fórmula (2.16) llega a ser

$$P(W) = \prod_{i=1}^{n} P(w_i \mid \Phi(w_1,...,w_{i-1})) \tag{2.17}$$

y el arte del modelado del lenguaje consiste en determinar la equivalencia de clasificación apropiada Φ y un método de estimación de probabilidades $P(w_i \mid \Phi(w_1,...,w_{i-1}))$.

Es un poco estresante que el modelo del lenguaje usado debiera depender sobre el uso en el cual el reconocedor será puesto. Por ejemplo, el dictado de reportes radiológicos requiere diferentes modelos del lenguaje que los escritos en revisiones de películas. Si se va a producir el texto, entonces el modelo del lenguaje se puede construir razonablemente procesando ejemplos de materiales escritos que correspondan. Así dependerá del texto solamente y en ninguna forma de la voz.

2.3.3 Búsqueda de la Hipótesis

Para encontrar la transcripción deseada $\hat{W}$ de los datos acústicos A por medio de la fórmula (2.6), debemos buscar en todas las posibles cadenas W para encontrar la que lo

maximice. Esta búsqueda no puede ser dirigida por fuerza bruta: porque el espacio de búsqueda de W es astronómicamente grande. Una búsqueda de la hipótesis es necesaria para que ni siquiera consideremos un número sobresaturado de posibles candidatos W y examinemos sólo aquellas cadenas en alguna forma por los datos acústicos A.

Una vez obtenida la hipótesis, podemos evaluarla con los datos reales que es nuestra referencia vía algún método estándar (NIST). Los modelos acústicos y del lenguaje pueden mejorar su desempeño utilizando métodos como adaptación y normalización, entre otros, los cuales veremos en el siguiente capítulo.

Hasta ahora se ha revisado parte de las bases o fundamentos de un reconocedor de voz así como algunos de los problemas que se presentan. En la siguiente sección veremos algunos métodos para mejorar el reconocimiento atacando diferentes puntos que pueden afectar el desempeño de cualquier sistema de reconocimiento de voz por ruido y diferencias entre los locutores mencionados anteriormente.