Proyecto Final de Carrera
Ingeniería Informática
Curso 2008-2009

Desarrollo de una prótesis de voz sintética para personas con discapacidad en el habla

Eduardo López Larraz

Septiembre de 2009

Director: Javier Mauricio Antelis Ortiz

Ponente: Javier Mínguez Zafra

Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior
Universidad de Zaragoza

A los que se han marchado antes de poderles dedicar este trabajo.

Agradecimientos

A todos aquellos que, de alguna manera, han hecho posible que este trabajo haya terminado de manera satisfactoria.

A Javier, Mauricio y Óscar, por haber sido la bibliografía mas importante de este proyecto.

A C&I, por estar siempre sentados ahí, por todos esos cafés y por resolverme tantas dudas.

A los buenos profesores que he tenido durante estos años, por haber dejado grandes conocimientos dentro de mi cabeza.

A los compañeros de clase y de prácticas, con los que tantas horas y quebraderos de cabeza he compartido a lo largo de esta travesía.

A mi familia, por animarme a trabajar día sí y día también. Con tantos ánimos, tenía que acabar algún día...

A todos mis amigos, por hacerme desconectar cuando las horas de trabajo saturaban mi mente.

A Carmela, por estar ahí por escucharme y animarme cuando los problemas más me sobrepasaban.

Muchas Gracias a todos.

Resumen

El objetivo de este proyecto de investigación es la realización de un estudio de viabilidad para el desarrollo de una prótesis de reconocimiento del habla mediante electromiografía. Esto es útil tanto para personas con alguna discapacidad en la producción de la voz como para aquellos que necesiten un sistema tradicional de reconocimiento del habla pero vayan a emplearlo en algún ambiente que dificulte el proceso, pongamos, por ejemplo, en una fábrica en la que el ruido es mucho mayor que la voz producida a una intensidad normal.

Las dos principales tecnologías empleadas para este trabajo son la electromiografía (EMG) y la inteligencia artificial. De la primera ha sido necesario establecer mecanismos tanto de obtención como de su posterior procesado; en el segundo campo se han empleado distintas máquinas de aprendizaje sobre las cuales se han realizado estudios y comparativas para establecer cuál se adecua más a nuestro problema.

La estructura del trabajo puede ser dividida en cinco grandes bloques. El primero de ellos es la documentación e investigación sobre las tecnologías a emplear y sobre sistemas similares desarrollados anteriormente. Cabe destacar que no se ha encontrado constancia de ningún proyecto de similares características realizado en castellano.

El siguiente paso corresponde al estudio fisiológico que, por una parte, tras investigar la naturaleza del habla y de los músculos faciales [1] [2] han permitido diseñar una configuración en la que se situarán los electrodos que captarán las señales EMG. Por otro lado, el estudio sobre la fonología española [3] tiene como resultado la definición de un vocabulario, compuesto por sílabas simples, que servirá para verificar el trabajo realizado.

En paralelo se ha ido trabajando en la construcción de las herramientas necesarias para el procesado de señal y en la realización de protocolos de experimentación.

También se ha trabajado con varias herramientas de clasificación y se han diseñado distintos esquemas con el objetivo de encontrar el método que permita distinguir entre el vocabulario de sílabas creado.

Por último, se han llevado a cabo una serie de sesiones de experimentación para adquirir señales EMG suficientes que permitan validar los bloques de procesamiento y clasificación mencionados.

Índice

Índice de figuras

Índice de tablas

1. Introducción

Uno de los tipos de interfaz entre persona y ordenador en el que más se ha investigado en los últimos años son los sistemas de reconocimiento del habla (en inglés automatic speech recognition, o ASR), que se basan en grabar las señales de voz producidas por una persona, procesarlas y clasificarlas para reconocer qué es lo que se ha dicho. Sin embargo estos interfaces presentan dos graves inconvenientes que pueden hacer inservible un sistema de este tipo:

- Si nos encontramos en un entorno ruidoso, como una fábrica o en una oficina donde pueden encontrarse más personas hablando, un ASR no resultará útil, ya que mezclará las señales de voz con el ruido y no conseguiremos que reconozca lo que deseamos.
- Para personas con cualquier tipo de discapacidad en el habla, estos sistemas tampoco resultan valiosos, dado que no serán capaces de identificar las señales de voz.

Para salvar estos inconvenientes se han empezado a desarrollar interfaces de reconocimiento del habla basados en electromiografía (EMG). Con esta tecnología no es necesario que el usuario produzca señales acústicas, ya que el simple movimiento de los músculos faciales produce un voltaje que puede ser detectado y medido con pequeños electrodos para después procesarlo en un computador.

El objetivo de este trabajo es realizar un estudio de viabilidad para el desarrollo de un prototipo de una prótesis que consiga reconocer patrones del habla sin necesidad de señales acústicas. Como punto de partida tomamos un proyecto llevado a cabo por una estudiante de ingeniería biomédica que realizó, en el Centro Politécnico Superior de Zaragoza, un estudio sobre el diseño de la prótesis y un prototipado del sistema con el que trató de reconocer las 5 vocales de la lengua española, pese a que los resultados obtenidos finalmente no fueron los esperados. Hasta el momento no hemos encontrado constancia de nadie más que haya realizado un sistema de este tipo en español, por tanto, podemos decir que seremos pioneros en este aspecto.

Los primeros meses de trabajo han consistido básicamente en la investigación sobre proyectos realizados por científicos de todo el mundo en este campo (ver [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]), que han desarrollado sistemas de reconocimiento del habla

que funcionan para las vocales o para unas pocas palabras. Han sido muchos los que han hecho progresos en estos sistemas, sin embargo, todavía no existe un reconocedor del habla efectivo que haya sido elaborado con EMG y que cubra todo un idioma. Por ello, plantaremos las bases para desarrollar uno, estableciendo como piezas elementales algunas sílabas simples del lenguaje castellano. Así, basándonos en los sonidos sencillos que componen las palabras, en un futuro, seremos capaces de construir cualquier combinación más compleja.

También ha habido que estudiar atentamente los músculos que componen la anatomía facial humana [2], [4], [14], [15]. Esto ha servido para conocer más detalladamente quiénes son los encargados de producir el movimiento que se lleva a cabo en la gesticulación de nuestros fonemas. Gracias a esto se ha establecido una configuración para determinar la localización de los electrodos, lo que permitirá conseguir unas señales más claras.

La primera distribución de electrodos estudiada fue de manera monopolar, colocando un electrodo sobre el vientre de cada músculo, con la tierra situada en la frente y la referencia en el lóbulo de la oreja. Esto se utilizó en un experimento de validación que, posteriormente, con los algoritmos de extracción de características y de clasificación ya implementados, nos permitió ver que los datos obtenidos eran excesivamente ruidosos y proporcionaban unos ratios de clasificación no demasiado buenos.

Por lo tanto, la configuración recomendada para los sensores es de manera bipolar [16], [17], [18], colocando dos electrodos de forma paralela a las fibras de cada músculo con respecto a un electrodo que actúe como tierra colocado en la frente y referenciados todos ellos al lóbulo de la oreja.

Para la obtención de las señales musculares de un usuario se ha diseñado un protocolo que emplea la aplicación BCI2000. Ésta se configura para recibir los datos de entrada provenientes de un amplificador al cual están conectados los electrodos que, en su otro extremo, están en contacto con los músculos faciales de una persona. La actividad muscular se graba en un fichero que separa internamente la información de cada uno de los electrodos o canales. El protocolo establece que por una pantalla se muestran sílabas de nuestro vocabulario durante un periodo de un segundo, intercaladas entre sí por un tiempo de reposo de, también, un segundo. La aplicación BCI2000 escribe en el mismo fichero de los datos un código numérico correspondiente a la sílaba mostrada durante los instante de tiempo en los que podía ser vista; y el código 0 durante el tiempo que no se muestran estímulos visuales. Con esto podemos conocer con exactitud las señales de todos los canales sincronizadas con el tiempo en el que se ha pronunciado cada una de las sílabas. Esto permitirá tomar el intervalo de tiempo en el que medir y extraer las características a las señales electromiográficas.

Los mecanismos de extracción de características se han desarrollado utilizando una serie de scripts programados en Matlab, que toman como entradas ficheros provenientes de BCI2000 y, conociendo el protocolo anterior, se valen de él para distinguir los momentos en los que se ha pronunciado una sílaba y aplicar las operaciones y transformaciones necesarias a las señales resultantes a fin de obtener una serie de valores que permitan

clasificarlas posteriormente. Estas características se almacenan en ficheros con un formato preestablecido para ser posteriormente utilizadas como valores de entrenamiento y clasificación.

Posteriormente, se ha empleado un software llamado Weka[1], que contiene una gran colección de herramientas de visualización y algoritmos de análisis y clasificación de datos. Esto es muy útil para observar las agrupaciones que puede presentarse en los distintos valores de las características o para seleccionar qué método de clasificación puede darnos mejores resultados para nuestro problema concreto.

Dentro de este programa, se han empleado como herramientas de clasificación un árbol de decisión, el método Naive Bayes y el metaclasificador Adaboost combinado con ambos. Todos ellos serán explicados en la sección 4.4.

Los resultados conseguidos para este estudio de viabilidad han superado las expectativas iniciales, ya que se ha conseguido desarrollar y validar una primera versión del prototipo, lográndose unos porcentajes de acierto en el reconocimiento de las vocales de un 80,2 % y un 70,93 % en el caso de la clasificación de 30 sílabas. Este resultado es equiparable al conseguido en otros trabajos similares. La mayoría de investigaciones consultadas corresponden a clasificación de las vocales, encontrándose resultados de reconocimiento correcto entre el 60 y 90 % de los casos. Para sistemas de reconocimiento de palabras, en [8] se distinguen 15 comandos diferentes al 74 %, mientras que en [6] lograron reconocer 6 palabras con un rendimiento superior al 90 %.

Para el desarrollo del proyecto el autor ha trabajado en un grupo multidisciplinar en el que se ha tenido que coordinar con profesionales en áreas de ingeniería electrónica, ingeniería informática e inteligencia artificial. Además, se ha colaborado con personal biomédico del Hospital Miguel Servet de Zaragoza, muy interesados en este trabajo de rehabilitación.

Este documento está estructurado en 6 capítulos, siendo este primero la introducción. En el capítulo 2 presenta un resumen del sistema global y el alcance de este proyecto concreto. El capítulo 3 muestra todos los detalles del estudio fisiológico realizado. El capítulo 4 describe todo el funcionamiento de los sistemas diseñados, tanto de interfaces, como de tratamiento de señales y clasificación. En el capítulo 5 se muestran los resultados más significativos que se han conseguido. Por último, en el 6º capítulo se presentan las conclusiones y la valoración sobre el trabajo realizado. Con respecto a la documentación extra, se han redactado 6 anexos ampliando los temas que se han considerado más relevantes. El anexo A describe el desarrollo del proyecto, mostrando el diagrama de Gantt y la distribución de tiempos empleada. En los anexos B y C se puede encontrar información acerca de la fisiología del habla y sobre la interfaz electromiográfica utilizada, respectivamente. El anexo D amplía la información sobre la extracción de características realizada a las señales EMG y el anexo E es un interesante resumen sobre las máquinas de aprendizaje y los problemas de clasificación. En último lugar, el anexo F adjunta todos los resultados

[1] http://www.cs.waikato.ac.nz/ml/weka/

de pruebas realizadas que, por falta de espacio, no han sido mostrados en el capítulo correspondiente de la memoria.

2. Resumen del sistema y alcance del proyecto

En este capítulo se muestra una visión global del funcionamiento del sistema en la sección 2.1 y el alcance del proyecto en la sección 2.2.

2.1. Resumen del sistema

El presente proyecto de investigación se enmarca dentro de un proyecto CICYT que lleva como título «Evaluación Biomédica de Robots de Ayuda a la Movilidad» financiado por el Ministerio de Ciencia y Tecnología. El autor se ha integrado dentro del equipo de investigación relacionado con este proyecto, el cual dispone de todas las infraestructuras necesarias para su soporte y realización. Además, ha existido una coordinación con el personal biomédico del Hospital Miguel Servet de Zaragoza, que han colaborado con gran interés en el primer prototipo de la prótesis del habla.

El desarrollo de la prótesis del habla mediante electromiografía es un proyecto muy ambicioso que requiere de varios pasos intermedios antes de conseguir una versión comercial. Para ello deben vencerse algunas barreras tecnológicas, como por ejemplo integrar los electrodos en una mascarilla, lo que facilitaría su colocación, o adaptar el software a un dispositivo móvil para hacer que el sistema sea portátil. Los proyectos que deberán realizarse para la construcción del sistema global son los siguientes:

1. Estudio de viabilidad que determine si es posible resolver el problema de clasificación a partir de las señales EMG obtenidas. Esto requiere una investigación sobre la fisiología muscular para extraer adecuadamente las señales electromiográficas, sobre la fonología española para empezar a distinguir entre fonemas representativos del idioma, hay que establecer un mecanismo para extraer una serie de características a las señales y, por último, hay que encontrar un clasificador que sea adecuado para el problema y proporcione unos resultados aceptables de clasificación.

2. Escalabilidad del sistema. Esto deberá realizarse en tres ámbitos distintos.

- En vocabulario. Para ello, deberá más minuciosamente el lenguaje para ampliar el conjunto de sílabas que se quieren reconocer.
- En tiempo. Lo cual requerirá de las pruebas necesarias para verificar si se pueden utilizar datos adquiridos en diferentes sesiones de experimentación, ya que pequeñas variaciones involuntarias en la localización de estos pueden causar que las señales sean tan distintas que para un clasificador sea imposible reconocerlas.
- En personas. Realizando experimentaciones con distintos sujetos y observando si para todos se obtienen unos resultados de reconocimiento similares.

Figura 2.1: Representación del sistema de reconocimiento del habla mediante EMG implementado en un dispositivo móvil.

3. Desarrollo de un mecanismo de reconocimiento en flujo, que permita reconocer concatenaciones de sílabas para formar palabras y, reconocimiento semántico que dé sentido a las frases, mejorando posibles errores del primer clasificador. Este paso y el

siguiente son muy similares a los que se realizarían con un sistema de reconocimiento de voz tradicional.

4. Miniaturización del sistema para hacerlo portable. Dado que lo que se quiere conseguir es una prótesis para que una persona con discapacidad en el habla se comunique correctamente, ésta debe acompañar a esa persona donde quiera que vaya, por tanto hay que adaptar el software de extracción de características y de clasificación para que funcione en un dispositivo móvil, como una PDA. También hay que conseguir que la obtención de las señales sea más sencilla, lo que se conseguiría con una mascarilla en la que se integren los electrodos y que permita que una persona se la coloque a sí misma con relativa facilidad.

La figura 2.1 muestra cómo quedaría en una persona la prótesis de reconocimiento del habla utilizando una máscara con los electrodos integrados, comunicada con un dispositivo móvil que, perfectamente, podría llevarse colgando en el cinturón para que, mientras el procesador ejecuta el software pertinente, sus altavoces reproducen las palabras pronunciadas.

2.2. Alcance del proyecto

El presente proyecto corresponde al paso 1 de los comentados en la sección anterior. Al tratarse más de un estudio de viabilidad que del desarrollo de un sistema, no va a proporcionar, a priori, una visión de aplicación global. Sin embargo, podemos tomarlo como un conjunto de subsistemas donde cada uno tiene un papel en la formación del prototipo. Los principales objetivos a cumplir en este proyecto son:

- **Realizar un estudio fisiológico** que, por una parte, permita conocer la anatomía facial de una persona, lo cual es imprescindible a la hora de colocar los electrodos de manera adecuada para extraer unas buenas señales electromiográficas. También se debe hacer un estudio del lenguaje castellano para conocer cómo se forman los distintos golpes de voz que forman las palabras y, con esto, establecer qué elementos tomar como unidades básicas.
- **Diseñar un sistema que sea capaz de discriminar entre un conjunto suficientemente representativo de sílabas**. Este diseño está compuesto por dos aspectos principales: (*i*) tratamiento de las señales para extraer de ellas una serie de valores o características (para ello se debe diseñar un mecanismo que automatice esa extracción), y (*ii*) diseño y elección del clasificador que mejor se adecua al problema y nos da unos buenos resultados. Dado que tenemos un problema de 30 clases, obtener en esta primera versión unos resultados de acierto superiores al 50 % sería aceptable, ya que un clasificador aleatorio proporcionaría un acierto del 3.3 %; alcanzar la barrera del 70 % sería un muy buen resultado.

- **Validar el sistema diseñado**. Para ello, deben programarse una serie de experimentos de adquisición de señales EMG, en los que se obtendrán una cantidad elevada de muestras de cada elemento del vocabulario para conseguir unos resultados que sean representativos. Con estas muestras, se entrenarán las máquinas de aprendizaje diseñadas y se realizarán los tests pertinentes que permitan verificar si la clasificación se lleva a cabo correctamente.

Estos objetivos se han cumplido de manera satisfactoria al haberse conseguido completar todas las tareas propuestas. En el anexo A, figura A.1 puede verse el diagrama de Gantt con la distribución del tiempo en el desarrollo del presente proyecto.

3. Estudio fisiológico

En este capítulo se muestra el estudio sobre las patologías que presentan los pacientes que pueden ser potenciales usuarios para la prótesis (sección 3.1), la anatomía de los músculos faciales, sobre los que se ha trabajado (sección 3.2) y el estudio realizado sobre la naturaleza del vocabulario escogido (sección 3.3).

3.1. Sintomatología

La voz es la máxima expresión del habla, una de las principales vías de comunicación empleadas por el ser humano. La ausencia de ésta por alguna malformación congénita o su pérdida a lo largo de la vida supone una barrera que, pese a no implicar necesariamente un riesgo vital, puede perjudicar en gran medida las relaciones personales. En el anexo B puede encontrarse más información acerca de la fisiología del habla y las patologías que pueden causar la pérdida de la voz.

Con el objetivo de mejorar la calidad de vida de personas con esa discapacidad nace la idea de desarrollar una prótesis del habla que, sin embargo, no sólo resultaría útil para esas personas; además podría ser ampliamente aplicada en entornos ruidosos en los que una comunicación telefónica clásica resulta prácticamente imposible.

Esta prótesis está, por tanto, dirigida a personas sanas o con ciertas discapacidades en el habla, producidas por malformaciones o lesiones en el aparato fonador. Para que un usuario pueda operar con el sistema debe conservar la mímica del habla, aunque sólo sea en la mitad de la cara, ya que sólo con eso y, gracias a una máquina de aprendizaje entrenada individualmente, se podría reconocer a qué corresponden las gesticulaciones realizadas.

La musculatura facial humana es un sistema muy complejo, pero normalmente es simétrico, por eso bastaría sólo con la movilidad en media cara para que el sistema pueda funcionar. Las figuras 3.1 y 3.2 muestran dos mapas con la anatomía muscular humana vistas de frente y de perfil.

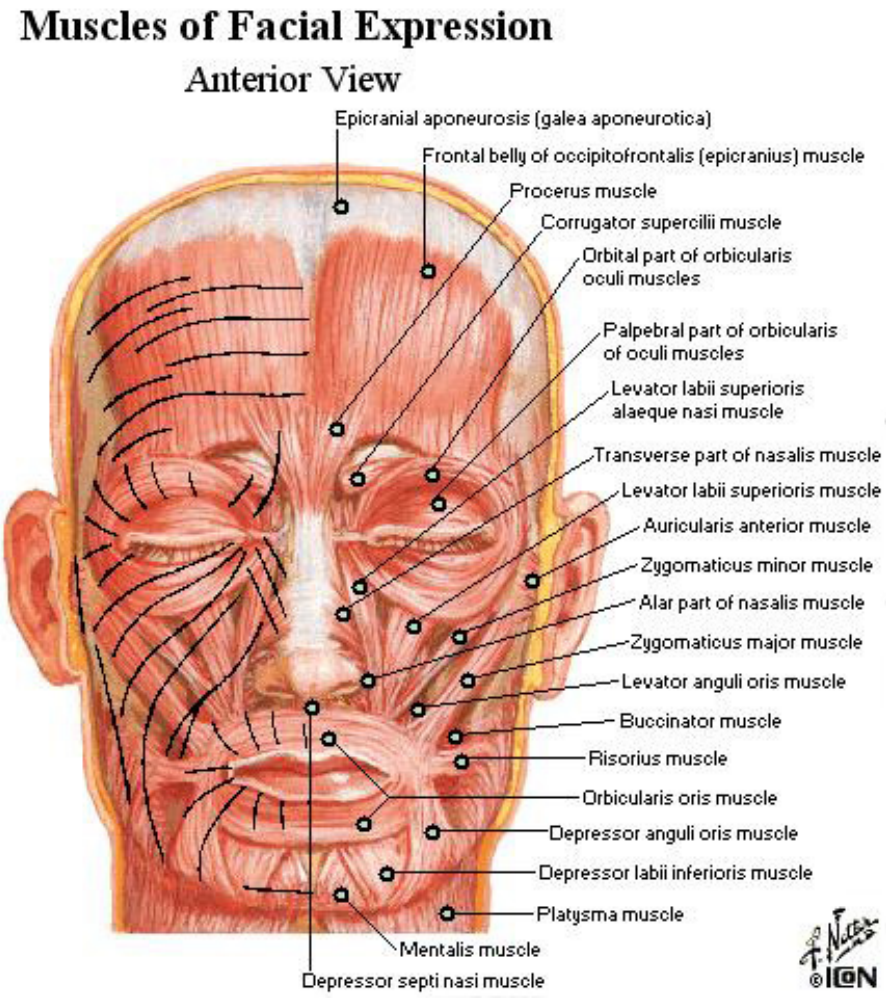


Figura 3.1: Vista anterior de la musculatura facial humana.

3.2. Estudio de la musculatura facial

El estudio de los músculos que componen la anatomía facial es un punto clave para la obtención de unas señales adecuadas que permitan una correcta clasificación posterior. Sobre su superficie deberemos colocar los electrodos para la adquisición de los datos, por tanto, una buena elección de los músculos, junto con una correcta colocación de los sensores nos proporcionará unas muestras suficientemente buenas con las que poder trabajar.

Pese a existir varios prototipos similares de reconocimiento del habla, en otros lenguajes, con EMG([4], [5], [6], [7], [8], [9], [10], [11], [12]), no existe ninguna convención sobre qué músculos medir y dónde se pueden obtener las señales óptimas para un buen reconocimiento.

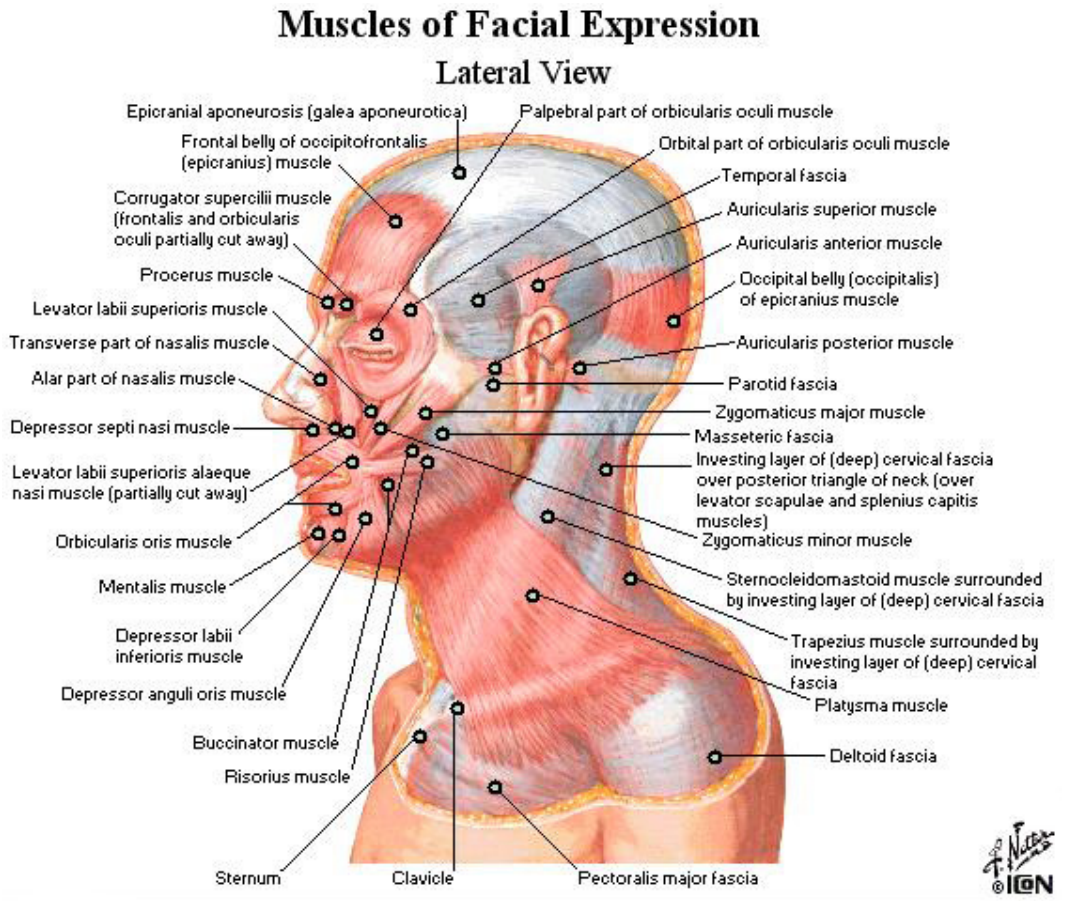


Figura 3.2: Vista lateral de la musculatura facial humana.

Por ello, tras varias pruebas se ha decidido emplear una estrategia en la que se colocarán 16 electrodos que, configurados de manera bipolar, nos proporcionarán 8 canales. La tierra será otro electrodo colocado en el centro de la frente y todos ellos tendrán como referencia el lóbulo de la oreja. Los músculos sobre los que se situarán los electrodos serán: *Levator labii superioris*, *Zygomaticus major*, *Orbicularis oris*, *Risorius*, *Depressor anguli oris*, *Depressor labii inferioris*, vientre anterior del músculo *Digastric*, y por último la lengua. Estos músculos se han escogido porque todos ellos han sido utilizados en alguno de los trabajos referenciados y, además están distribuidos por toda la cara, no restringiendo así la colocación a una zona determinada de ésta. En la figura 3.3 podemos ver esta configuración.

3.3. Estudio del vocabulario

Dado un claro objetivo, como es el de diseñar una prótesis de reconocimiento del habla, necesitamos estudiar la naturaleza de ésta para comprender cómo se forma y qué manera

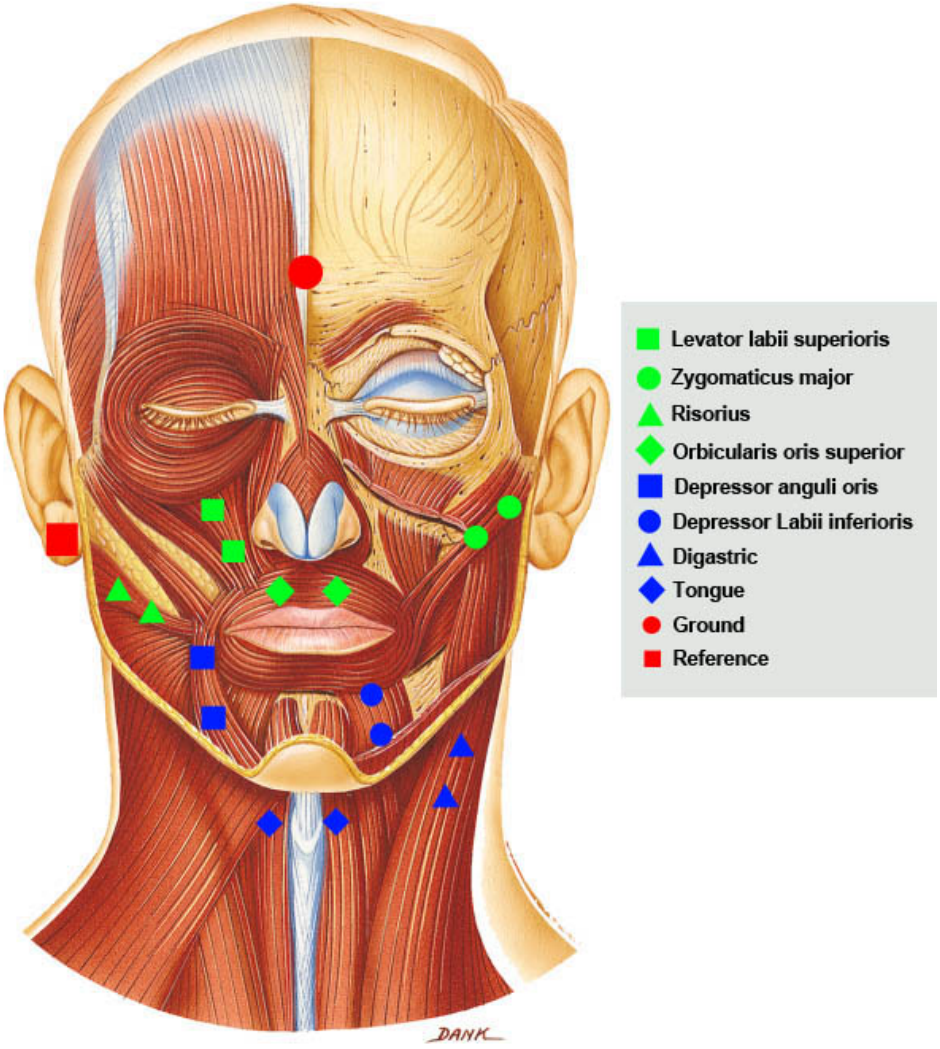


Figura 3.3: Mapa donde podemos observar la musculatura facial y donde aparecen, con diferentes formas y colores, los lugares donde se colocarán los electrodos.

será la idónea para desarrollar el mecanismo reconocedor. Pese a que el propósito final es el de abarcar todo el lenguaje castellano, el primer paso antes de reconocer sus palabras es tratar de reconocer las sílabas que las componen. Así, siguiendo este orden lógico, primero se reconocerán estas unidades básicas del habla, posteriormente serán agrupadas en palabras y, más adelante, se buscará obtener un reconocimiento semántico que dote de sentido a las frases formadas.

Una vez en este punto debemos definir cuáles son las sílabas que queremos incluir en nuestro sistema. Deberán ser un subconjunto de las sílabas más simples, compuestas por una consonante seguida de una vocal y buscaremos las que, de alguna manera, sean suficientemente representativas.

Por un lado vamos a tomar las sílabas formadas únicamente por sonidos vocálicos, es

decir, las cinco vocales. Con respecto a las que poseen sonidos consonánticos, podemos clasificar este tipo de sílabas según su punto de articulación en cinco grupos principales:

1. **Labiales**
   Son los sonidos que obedecen al brusco despegue de los labios, con el objetivo de expulsar rápidamente el aire contenido en la cavidad bucal. Las consonantes representativas de este grupo son la **b**, la **m** y la **p**; componiendo los sonidos labiales al acompañarse de las cinco vocales.

2. **Dentales**
   Para una correcta pronunciación de los fonemas pertenecientes a este grupo, empleamos la lengua con punto de apoyo en los dientes superiores y, como en las labiales, abrimos la boca con rapidez. El grupo de consonantes que integran este grupo son la **d**, la **t**, la **z** y la **c** en su sonido [**ce**], [**ci**].

3. **Palatales** o **paladiales**
   En este caso, la articulación de los sonidos es similar al grupo anterior, sin embargo, es la parte central de la lengua la que empleamos como válvula y la parte superior del paladar la que tomamos como punto de apoyo, justo por encima de los alvéolos de los dientes superiores. Conseguimos la pronunciación de estas sílabas con los sonidos consonánticos **ll**, **ñ**, **ch** e **y**.

4. **Velares** o **guturales**
   La parte posterior de la lengua, con una separación brusca del velo del paladar a la vez que se abre la boca, permite la expulsión repentina del aire y produce este tipo de sonidos. Integran este grupo todas las integrantes del sonido [**k**], la **g** (como [**ga**], [**gue**], [**gui**], [**go**], [**gu**]), y la **j**.

5. **Alveolares**
   En la producción de estos sonidos, la lengua se apoya (a veces de manera intermitente) sobre los alvéolos dentales superiores. Se componen por las letras **l**, **r**, **s** y **n**, combinadas, como siempre, con las cinco vocales.

Para escoger un conjunto de sílabas representativo se han tomado dos consonantes de cada uno de estos grupos combinada con las vocales, además de las cinco vocales por separado; lo que hace un total de 55 clases. Las consonantes elegidas han sido: **P**, **M**, **T**, **Z**, **Y**, **CH**, **K**, **J**, **L** y **S**, por ser representativas dentro de sus grupos y permitir después construir palabras sencillas mediante su combinación. A cada una de estas clases se le ha asignado un código numérico que va del 1 al 55.

Sin embargo, para la experimentación final se decidió eliminar una consonante de cada grupo, dado que era inviable, por cuestiones temporales, adquirir una cantidad mínima de 100 muestras por cada sílaba en una sola sesión. Como se comentó en la sección 2.1, la utilización de datos pertenecientes a diferentes sesiones de experimentación está en el alcance del segundo proyecto para el desarrollo de la prótesis y se consideró que sería

preferible reducir el vocabulario para adquirir un número elevado de muestras de cada sílaba. Por tanto, finalmente el vocabulario tomado lo forman 30 clases: las cinco vocales y la combinación de éstas con las consonantes **P**, **T**, **Y**, **K** y **L** (tabla 3.1).

Vocales	A	E	I	O	U
Labiales	PA	PE	PI	PO	PU
Dentales	TA	TE	TI	TO	TU
Palatales	YA	YE	YI	YO	YU
Velares	KA	KE	KI	KO	KU
Alveolares	LA	LE	LI	LO	LU

Tabla 3.1: Conjunto de sílabas que formarán el vocabulario a reconocer.

4. Diseño del sistema

En este capítulo se expone el diseño de los sistemas que integran el proyecto en la sección 4.1. La sección 4.2 describe el sistema de interacción entre persona y ordenador, la 4.3 el sistema de tratamiento de señales y, por último, la sección 4.4 las máquinas de aprendizaje exploradas.

4.1. Estructura de los sistemas

Teniendo en mente el objetivo de realizar un estudio de viabilidad para el desarrollo de una prótesis del habla, se va a diseñar un primer prototipo que trate de reconocer un subconjunto de sílabas simples. Para ello se dividirá el trabajo en una serie de bloques tal y como muestra la figura 4.1.

El primer bloque es el encargado de permitir la interacción entre persona y ordenador para captar de un usuario las señales EMG. El sistema de procesado de señales, las transformará en una serie de valores correspondientes al resultado de aplicar operaciones o transformaciones sobre las mismas. Por último una máquina de aprendizaje será diseñada y entrenada con un número suficientemente amplio de muestras para cada clase, con el objetivo de que, después, al introducir nuevos ejemplos sea capaz de clasificarlos correctamente.

4.2. Interfaz Hombre-Máquina

En esta sección se exponen todos los componentes que integran el sistema de interacción hombre-máquina. En el apartado 4.2.1 se describirá el protocolo fisiológico diseñado para la obtención de las señales EMG. Seguidamente, en el apartado 4.2.2 se mostrará el sistema encargado de la adquisición, realizar el primer tratamiento y almacenamiento de las señales. Por último se describirá la interfaz visual, creada para conseguir una adquisición controlada y ordenada de los datos, en el apartado 4.2.3.

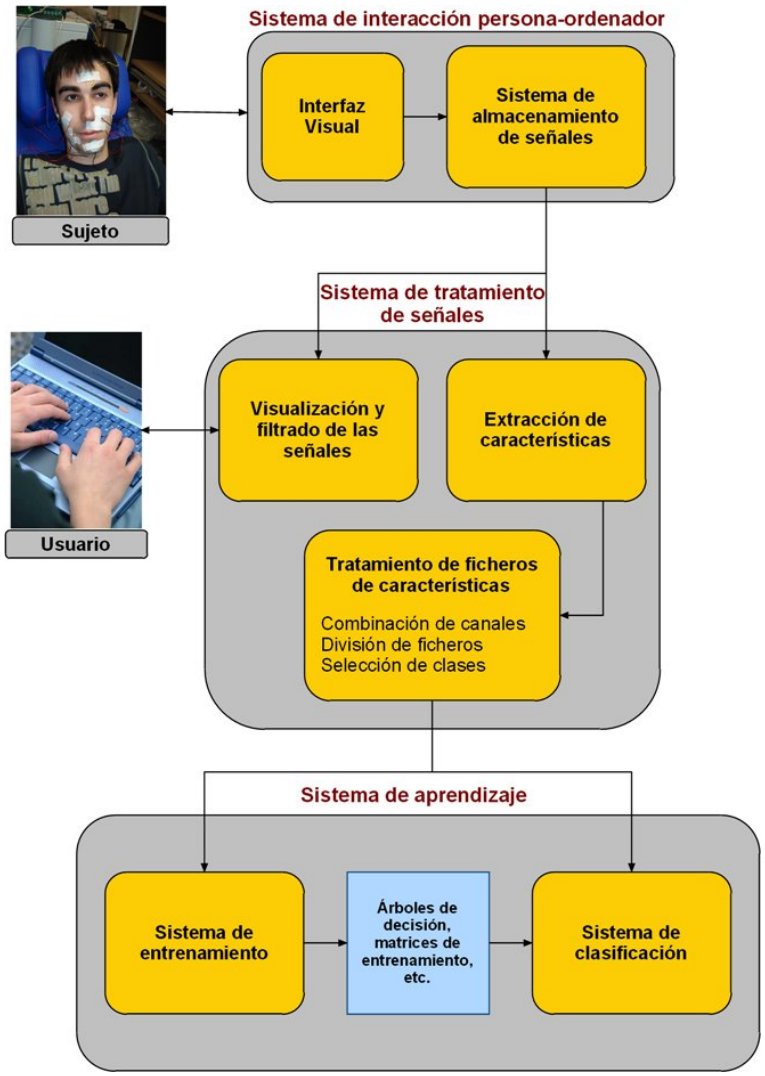


Figura 4.1: Esta figura muestra cómo están formados y la interacción de los tres sistemas principales.

4.2.1. Protocolo fisiológico

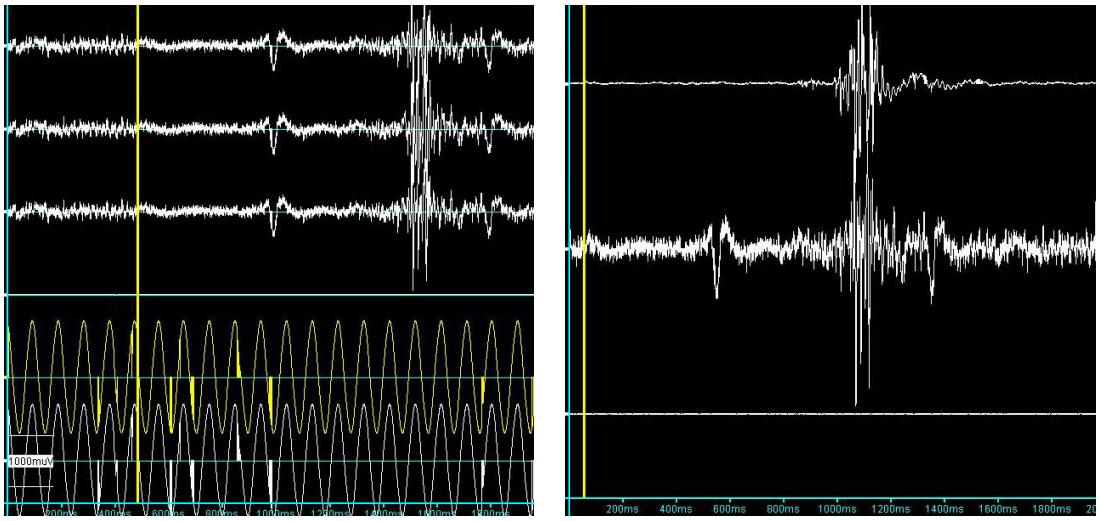
Existen, fundamentalmente, dos formas de adquisición de EMG: las técnicas invasivas y las no invasivas. Las primeras requieren de electrodos con forma de aguja, que atraviesen la piel, para estar en contacto directo con el músculo que se desea medir. Las técnicas no invasivas, a las que también podemos referirnos como SEMG[1], se diferencian de las primeras en que la adquisición de los datos se realiza al poner en contacto el sensor con la piel. Ambas técnicas tienen sus ventajas e inconvenientes. Las primeras permiten una mayor selectividad de las fibras musculares que se desean medir, además proporcionan una señal mucho más limpia, dado que las interferencias que producen los músculos próximos a la posición del electrodo son mínimas; por contra, la inserción de este tipo de sensores es

[1]Surface electromyography

un proceso más delicado y peligroso que requiere personal con experiencia, también juega en su contra el hecho de tener que clavar agujas en las fibras musculares del paciente, ya que hace que ciertos movimientos puedan verse dificultados. Por otro lado, las técnicas no invasivas nos aseguran una adquisición más segura y fácil de configurar, sin embargo, presentan como inconveniente lo que son las ventajas de los métodos invasivos: la escasa selectividad y las interferencias entre músculos próximos.

Para nuestro sistema emplearemos electrodos superficiales; nos proporcionarán peor calidad de señal, pero compensan con la mayor seguridad. De cualquier manera, habría que explorar si la mejora propiciada por los electrodos de aguja es tal, que en una posible futura prótesis comercial compensase el utilizarlos.

Los músculos que vamos a medir son ocho: *Levator labii superioris*, *Zygomaticus major*, *Orbicularis oris*, *Risorius*, *Depressor anguli oris*, *Depressor labii inferioris*, vientre anterior del músculo *Digastric*, y la lengua. Éste último no se medirá sobre su superficie, sino colocando los electrodos a unos 3 centímetros debajo de la barbilla. Tenemos dos posibilidades para adquirir el SEMG: de manera monopolar o bipolar (figura 4.2). La primera configuración requiere un electrodo por cada músculo, colocando uno más que haga de tierra y otro de referencia. La segunda emplea dos electrodos sobre cada posición que se quiera medir, siendo, de igual modo, necesario utilizar tierra y referencia. La diferencia radica en que, con una configuración bipolar, la señal resultante nos la da la resta de las obtenidas por los dos electrodos, eliminando así ruido como el que producen los parpadeos de los ojos o el mismo latido del corazón. El inconveniente que tiene es que se emplean casi el doble de electrodos.



(a) (b)

Figura 4.2: Estas capturas pertenecen al experimento de validación donde se comprueba que las señales del gráfico (b) son el resultado de restar por parejas las del gráfico (a). Las señales 1 y 2 de (a) pertenecen a dos electrodos colocados sobre un mismo músculo a una distancia de 1 centímetro, la 3 es una copia de la 1, la 4 corresponde al electrodo de tierra y las dos últimas son idénticas y corresponden a un generador de señales sinusoidales, lo que hace que su resta sea una señal con amplitud cero.

La primera distribución de los electrodos que se propuso fue de manera monopolar. En un experimento de prueba se recogieron datos para una primera validación de los algoritmos de extracción de características implementados. Los resultados obtenidos en los primeros tests de clasificación no fueron muy alentadores, ya que los ratios de acierto obtenidos fueron muy bajos. Así que se decidió preparar una configuración bipolar para los siguientes experimentos de toma de datos, con los que los resultados fueron notablemente mejores. Se tuvo que realizar una adaptación del software de adquisición de datos para poder configurarlo de manera bipolar, ya que por defecto no teníamos esa posibilidad.

La infraestructura que será necesaria finalmente para la adquisición de señales EMG con una configuración bipolar será un conjunto de 18 electrodos, 16 para la recogida de las señales, uno para tierra y otro de referencia. Por otra parte, se necesita un amplificador que recibe las señales de los electrodos y se conecta con un computador mediante un cable USB.

Con el fin de obtener una señal lo más limpia posible, debe establecerse un protocolo de montaje que, además, permita la repetibilidad de cada sesión de experimentación, lo que minimizará los errores por la variación en la posición de los sensores. Antes de comenzar, hay que advertir que, si el sujeto sobre el que se van a colocar los electrodos es un varón, deberá afeitarse correctamente la cara, ya que el vello interfiere negativamente en la adquisición del EMG y, por muy meticulosamente que se sigan los siguientes pasos, la señal obtenida finalmente puede contener tanto ruido que la haga inútil.

Dicho esto, el primer paso, una vez que esté colocada la persona en un asiento cómodo, se le debe limpiar la superficie de la piel, ya sea con algún tipo de gel abrasivo o simplemente con un algodón mojado en alcohol. Esto eliminará partículas como restos de piel muerta, que pueden interferir negativamente en la adquisición. En los experimentos realizados, la técnica empleada ha sido la de limpiar la piel con alcohol, teniendo siempre extremo cuidado para evitar el contacto con los ojos, la nariz y la boca. Una vez hecho esto, el siguiente paso es ya la colocación de los electrodos. En la figura 3.3 se muestra la distribución de manera bipolar de estos, la colocación de cada par se hará siempre buscando el centro de los vientres musculares y colocándolos de manera paralela a las fibras.

Al terminar de colocar todos los electrodos, deben comprobarse las impedancias de cada uno de ellos, para comprobar cómo de buena es la conductividad en los puntos donde están colocados. Son aceptables valores menores a los 10kΩ. En caso de que alguno de ellos tenga una impedancia mayor, deberá revisarse su colocación y volverse a limpiar la piel que estaba en contacto con él. Cuando todo esto esté listo, ya podemos comenzar a registrar la actividad muscular.

4.2.2. Adquisición de datos

Para el sistema que hace de interfaz entre persona y ordenador se ha empleado una plataforma de código abierto llamada BCI2000 [19]. Este entorno ha sido desarrollado para la adquisición de señales EEG[2], usadas ampliamente para interfaces cerebro-ordenador (BCI[3] [20]), sin embargo, es totalmente válido para nuestro propósito, ya que las señales EMG son también señales fisiológicas y su modo de obtención y tratamiento es muy similar. También, el hecho de disponer de la plataforma BCI2000 totalmente operativa, por ser ampliamente utilizada por el equipo BCI del Centro Politécnico Superior, garantizaba asesoramiento ante cualquier problema que pudiese haber y finalmente, decantó la balanza hacia su utilización.

BCI2000 cuenta con un módulo de sincronización y tres módulos principales: de adquisición, procesado y aplicación. Para este proyecto únicamente se han empleado el de adquisición, que será explicado en este apartado, y el de aplicación, del que se hablará en el apartado 4.2.3. El módulo de procesado no se ha utilizado, ya que todos los procedimientos para el tratamiento de señal se implementarán aparte, mediante scripts en Matlab.

El módulo de adquisición de BCI2000 ha permitido configurar completamente los parámetros del amplificador utilizado para recoger las señales EMG. La frecuencia de muestreo empleada para registrar los datos ha sido de 2.400 Hz. Las señales adquiridas por los electrodos han sido filtradas mediante un filtro paso-banda, implementado en el hardware del amplificador, de 5 a 500 Hz, ya que es en esa banda de frecuencias donde se halla la información más importante. También se ha empleado un *notch-filter*, para evitar el rango de frecuencias de 48 a 52 Hz, debido a que en esa banda se produce ruido por la interacción con la instalación eléctrica del edificio.

4.2.3. Interfaz visual

Como se ha mencionado en el apartado anterior, se ha utilizado el módulo de aplicación de BCI2000 para diseñar un protocolo que permita la adquisición de datos de manera controlada. La aplicación utilizada se llama *StimulusPresentation* y su función es la de mostrar una serie de estímulos visuales o sonoros en instantes de tiempo indicados. Esta aplicación dispone de una amplia variedad de estímulos predefinidos, sin embargo, se ha adaptado para este proyecto añadiendo, por cada uno de los 30 componentes de nuestro vocabulario, mostrado en la sección 3.3, una imagen que contenga la letra o sílaba correspondiente. En la figura 4.3 vemos el estímulo creado para la sílaba *PA*.

La adquisición de los datos se realizó en sesiones de 25 ó 50 muestras de distintas sílabas que la aplicación almacena en ficheros, separando internamente la información de cada uno de los electrodos. La metodología diseñada establecía que cada sesión comenzaba

[2]Electroencephalography
[3]Brain Computer Interfaces

Figura 4.3: La imagen muestra el estímulo creado para indicar la sílaba *PA*, se creó uno por cada una de las 55 componentes del vocabulario y se mostraban de forma aleatoria al sujeto para que las pronunciase y poder así registrar su actividad muscular.

con una periodo de reposo de 5 segundos, para que el sujeto relaje la musculatura, después comienza la exposición visual de estímulos. Cada estímulo aparece en la pantalla durante un segundo y se intercala del siguiente por un tiempo de reposo, también de un segundo. El almacenamiento de las señales está totalmente sincronizado con la aparición de los estímulos por la pantalla. Al muestrear las señales a 2.400 Hz, tendremos entonces que cada señal perteneciente a una pronunciación, estará compuesta de 2.400 valores.

Para evitar la propensión del usuario a repetir siempre el mismo gesto en la pronunciación, se mostraban al azar los estímulos, teniendo, así, en un mismo fichero datos de distintas sílabas. La aplicación de BCI2000 escribe en el mismo fichero de los datos el código numérico correspondiente a la sílaba mostrada durante los instantes de tiempo en los que podía ser vista; y el código 0 durante el tiempo que no se muestran estímulos visuales. Con esto podemos conocer con exactitud las señales de todos los canales sincronizadas con el tiempo en el que se ha pronunciado cada una de las sílabas. Esto permitirá tomar el intervalo de tiempo en el que medir y extraer las características a las señales electromiográficas.

4.3. Tratamiento de señales

En esta sección se describen las herramientas desarrolladas para realizar el procesado de las señales adquiridas. En el apartado 4.3.1 se detallara la primera de ellas, que permite la visualización y procesado de las señales EMG. Por otra parte, en el apartado 4.3.2 se mostrará la otra aplicación, que toma como entrada los ficheros de características generados por la primera y realiza sobre ellos las modificaciones necesarias para su posterior ejecución con las herramientas de clasificación.

4.3.1. Visualización y procesado de señales

Esta herramienta ha sido desarrollada para automatizar, en la mayor medida posible, la extracción de características a las señales EMG. Con ella pueden verse también las gráficas para detectar así posibles anomalías en éstas. En la figura 4.4 se muestra la pantalla principal de la aplicación. En los siguientes subapartados se muestra en mayor detalle cada uno de sus módulos.

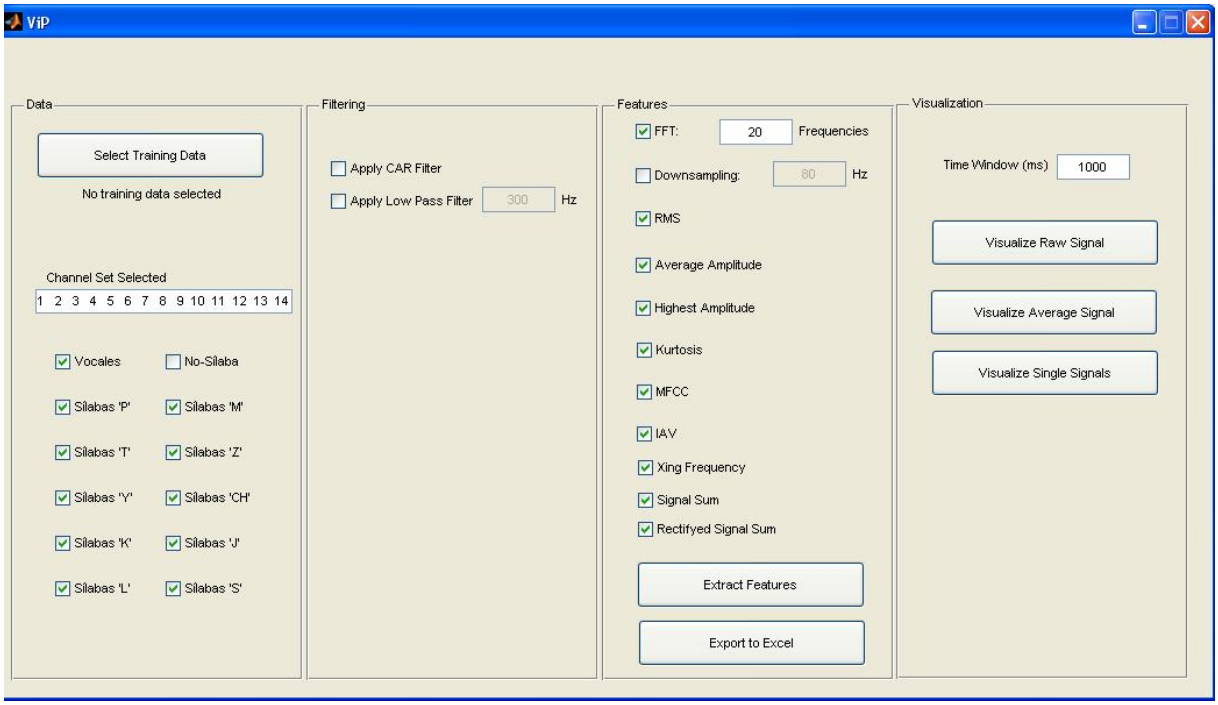


Figura 4.4: Captura de la pantalla principal de la aplicación creada para la visualización y el procesado de las señales EMG. El primer bloque corresponde a la selección de los datos, el segundo al filtrado, el tercero a la selección de características y el cuarto a la visualización de las señales.

4.3.1.1. Selección de datos

Desde este módulo se seleccionan los datos de entrada que después se procesarán. Permite seleccionar varios ficheros, con extensión *.dat*, que contengan datos de señales EMG. También posibilita seleccionar qué canales se desean leer, ya que podemos tener algún canal ruidoso o poco significativo que queramos excluir y sobre el que no queremos extraer características. Debido a la configuración bipolar escogida finalmente, el conjunto de canales seleccionados por defecto es de 1 a 16, ya que se emplean 16 electrodos; si quisiéramos quitar un canal bipolar deberíamos borrar de la lista los dos canales monopolares que lo componen[4]. Por último, nos permite seleccionar los grupos de sílabas con los

[4]Para quitar el canal bipolar 1, deberíamos borrar los canales monopolares 1 y 2; para el 2, deberíamos borrar 3 y 4; etc.

que queremos trabajar, pudiendo seleccionar sólo las vocales, las sílabas pertenecientes al grupo de la *P*, etc. Esto es útil porque podemos encontrar archivos que contengan datos de sílabas con las que no queremos trabajar y es una forma útil de filtrarlas.

4.3.1.2. Filtrado

Este módulo nos permite realizar dos tipos de filtrado: *CAR filter*, que elimina la media de la señal; y filtros paso bajas con valor configurable de paso. El hecho de que no se haya implementado un filtro paso altas es porque en las señales con las que trabajamos, la mayor parte de la información está contenida en las frecuencias bajas, siendo las altas frecuencias ruido en la mayoría de los casos, con lo que eliminaríamos casi toda la información útil.

Pese a haberse probado la clasificación con distintas configuraciones de los filtros mencionados, los resultados siempre han sido mejores al extraer las características con la señal sin filtrar, ya que, aunque se tomaran valores altos para el filtro paso bajas, siempre se eliminaba algo de información que, finalmente, resultaba valiosa en vista de los resultados que se obtuvieron. Por tanto todos los resultados expuestos en el capítulo 5 son de clasificaciones con las características extraídas a las señales sin filtrar.

4.3.1.3. Selección de características

Éste es el bloque más importante de la aplicación, ya que es donde se ejecuta todo el grueso de procesado de la señal. Además es la parte que más esfuerzo ha requerido en lo que se refiere a programación. Este módulo debe tomar como entrada las señales seleccionadas, en crudo o filtradas según se haya seleccionado, y aplicar sobre ellas una serie de operaciones y transformaciones para obtener unos valores que serán las características con las que se trabajará más adelante.

Las características que se han implementado son las que se listan a continuación. En el anexo D se explica con detalle qué es cada una de ellas.

- **Fast Fourier Transform (FFT)**
- **Downsampling de la señal**
- **Root Mean Square (RMS)**
- **Amplitud media**
- **Amplitud máxima**
- **Kurtosis**
- **Mel Frequency Cepstral Coefficients (MFCC)**

- **Integrated Absolute Value (IAV)**
- **Zero Crossing**
- **Suma de la señal**
- **Suma de la señal rectificada**

Pulsando el botón *Extract Features*, el programa aplica, para cada canal de cada señal, todas las características seleccionadas, y genera un fichero de texto en el que las almacena. El botón *Export to Excel* realiza las mismas operaciones, pero almacena los resultados en hojas de cálculo para una mejor inspección.

4.3.1.4. Visualización de señales

En este bloque disponemos de tres opciones. La primera de ellas permite dibujar toda la señal correspondiente a una sesión de adquisición. El resultado (ejemplo en la figura 4.5) son una serie de gráficas; la primera corresponde a la secuencia de estímulos mostrados, como ya se explicó en la sección 4.2.3, el programa escribía un 0 si no hay estímulo y el código de sílaba correspondiente al mostrar uno. Las siguientes gráficas corresponden a los canales que hayan sido escogidos en el módulo de selección de datos.

La segunda opción dibuja la señal media de todas las muestras contenidas en un fichero de datos. Con esto se puede observar la señal *tipo* de cada sílaba para compararla después con cada muestra por separado. La tercera separa cada muestra de una sesión de adquisición y las dibuja en una ventana separada.

4.3.2. Tratamiento de ficheros de características

Esta aplicación es mucho más sencilla que la anterior y fue creada para automatizar las distintas transformaciones a realizar sobre los ficheros de texto generados previamente. Esto se debe a que el formato en el que se generan estos ficheros es diferente a los que se utilizan para realizar las clasificaciones y adaptar los formatos manualmente es algo mucho más costoso. La figura 4.6 corresponde a la pantalla principal de esta herramienta.

4.3.2.1. Combinación de canales

La primera operación que nos permite realizar esta herramienta es el concatenar ficheros de características correspondientes a los distintos canales de unas mismas señales. La salida de la aplicación comentada en el apartado 4.3.1 son una serie de ficheros de texto (tantos como canales hayan sido seleccionados) que contienen, en cada línea, las

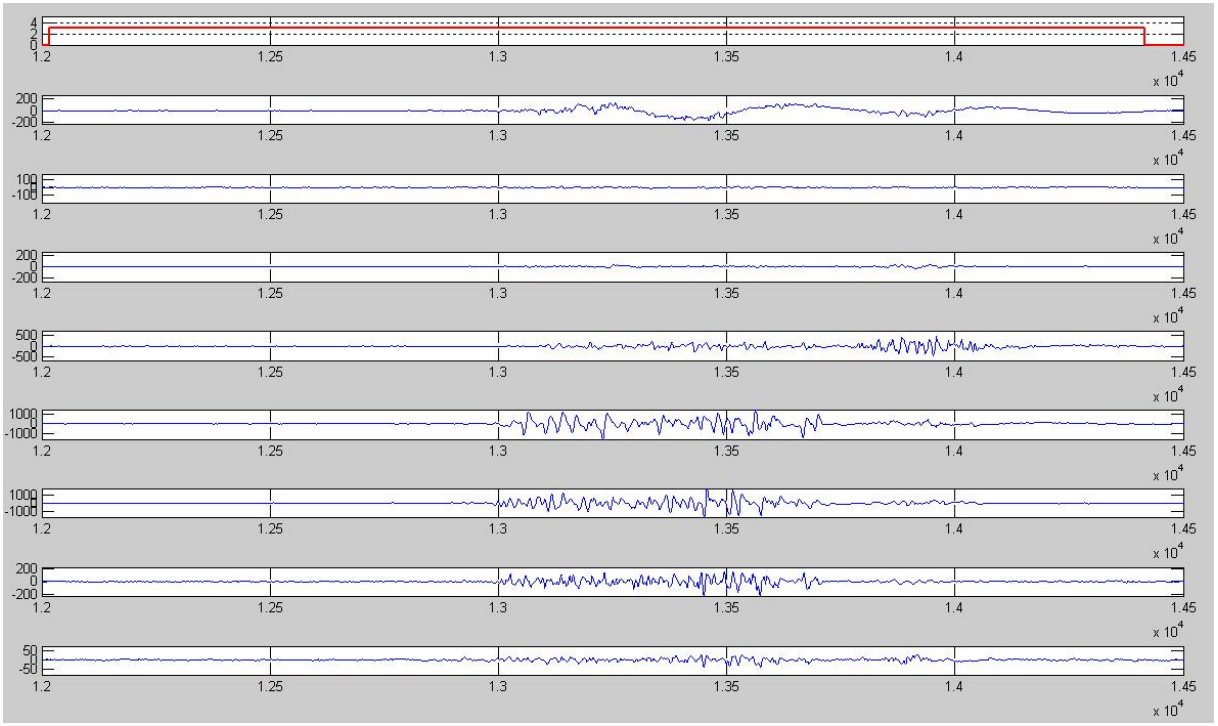


Figura 4.5: Gráficas correspondientes a la pronunciación de una *I*. Vemos que la primera tiene el valor 3 (código asignado a la *I*) durante un intervalo de tiempo y, durante ese tiempo, se observa una actividad notable en algunos de los canales.

características correspondientes a una muestra distinta de una sílaba. Así la primera fila del primer fichero de texto serán las características del primer canal de una muestra recogida de alguna sílaba; para el segundo fichero, la primera fila corresponderá al segundo canal de la misma muestra y así sucesivamente. Por tanto, si queremos trabajar con todos los canales juntos, debemos concatenar en una misma línea las características correspondientes a los 8 canales de cada muestra.

El hecho de concatenar todos los canales viene motivado porque ante un problema en el que se tienen varios, juntar todos ellos en una única señal permite contener toda la información disponible en una especie de canal virtual que engloba a todos. Juntar esos 8 ficheros manualmente es un trabajo realmente tedioso, por eso se optó por automatizar la tarea.

4.3.2.2. División de ficheros de datos

Cuando tenemos un fichero de características como los mencionados anteriormente y queremos probar algún algoritmo de clasificación, nos conviene utilizar un subconjunto de los datos para entrenar el sistema y el resto para testearla. Con esta opción basta con introducir el número de muestras que hay de cada clase y el porcentaje de entrenamiento-clasificación deseado para que genere dos ficheros divididos al azar con las proporciones

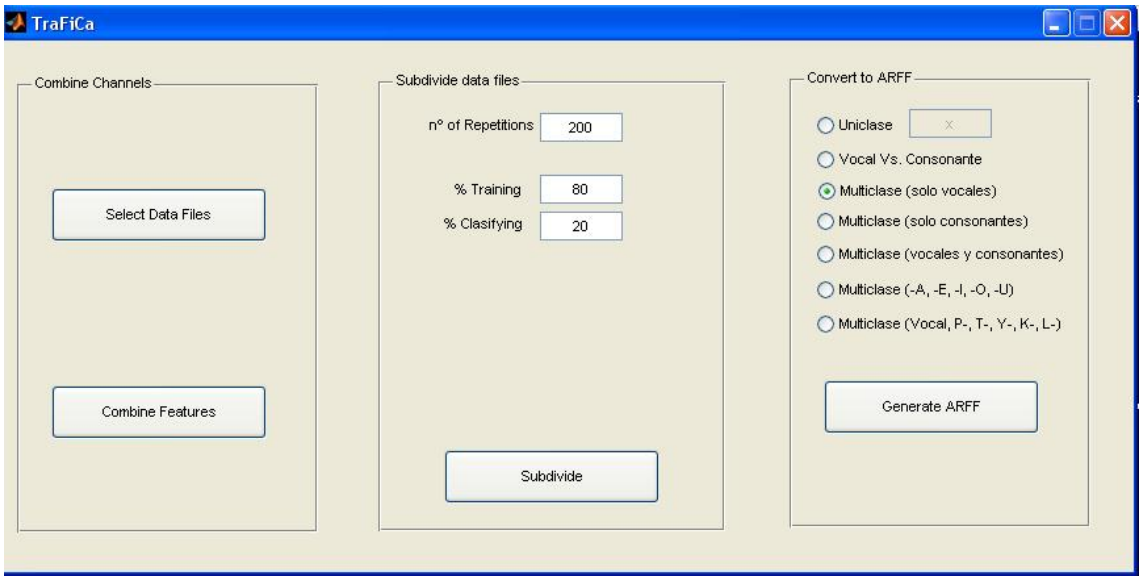


Figura 4.6: Captura de la pantalla principal de la aplicación de tratamiento de ficheros de características. El primer bloque es para la combinación de ficheros de los distintos canales, el segundo para subdividir ficheros en subconjuntos de entrenamiento y clasificación, el tercer bloque proporciona la funcionalidad de transformar ficheros de características al formato ARFF.

indicadas.

4.3.2.3. Conversión a ficheros ARFF

La herramienta Weka, usada para la gran mayoría de pruebas de clasificación realizadas, requiere que los datos de entrada estén en un formato específico, con extensión *.arff*. En el anexo E.7 se profundiza más en este formato. Con el último módulo de esta herramienta se puede transformar los ficheros de texto generados anteriormente al nuevo formato, presentando distintas opciones sobre las clases que se desean incluir en el fichero resultante.

4.4. Sistema de aprendizaje

Las máquinas de aprendizaje son una solución al problema clásico de reconocimiento de patrones. En esta sección se tratará todo lo relacionado con las herramientas de aprendizaje utilizadas. El apartado 4.4.1 es una introducción al problema de clasificación de patrones, el apartado 4.4.2 muestra los clasificadores estudiados y, por último, en el apartado 4.4.3 se describe el software Weka, que ha sido el interfaz de alto nivel con el que se han manejado los clasificadores mencionados.

4.4.1. Introducción al problema de clasificación

El reconocimiento de patrones es una disciplina científica en la que el objetivo es la clasificación de objetos dentro de un número de categorías o clases [21]. Antes de abordar el problema en sí, deben definirse unos conceptos que se utilizarán ampliamente al hablar de clasificación.

- **Clases.** Las clases son los conceptos o grupos en los que queremos ordenar todos los datos pertenecientes al *universo* que forma el problema de clasificación. En el caso de este proyecto tenemos 30 clases, que son las 5 vocales, más las 25 sílabas resultado de concatenar las consonantes *P*, *T*, *Y*, *K* y *L* con las vocales.

- **Instancias** o **muestras**. Las instancias se refieren a cada uno de los ejemplos concretos que tenemos pertenecientes una determinada clase. Son, realmente, los datos que queremos clasificar o agrupar. Para este problema, cada una de las veces que se adquiere la señal de una sílaba, es una nueva muestra. La figura 4.5 corresponde a una muestra tomada de la *I*.

- **Atributos**. Cada instancia individual introducida en una máquina de aprendizaje está compuesta por una serie de valores o atributos. Estos atributos pueden ser numéricos o nominales. En nuestro *universo*, todas las características o atributos son valores numéricos calculados con operaciones o transformadas de la señal (apartado 4.3.1.3). Sin embargo el atributo *Clase*, pese a ser representado con un número, es considerado nominal, ya que estos números son valores predefinidos en un espacio finito.

La idea básica para resolver este problema es utilizar una máquina que trate de *aprender*, a partir de una serie de muestras, a distinguir entre las distintas clases. Para ello, se le deben proporcionar una serie de instancias y, dependiendo del método de clasificación, buscará los mejores atributos de una u otra forma para identificar cada uno de los distintos grupos.

Una vez la máquina está *entrenada* se le pueden introducir nuevas instancias, que clasificará con un porcentaje de acierto mayor o menor dependiendo de distintos aspectos como pueden ser el tipo de clasificador empleado, el número de ejemplos proporcionados para el entrenamiento, la separabilidad de las clases, etc. Que las clases sean poco separables quiere decir que datos que, teóricamente, corresponden a distintos grupos, poseen atributos con valores muy semejantes, lo que puede causar que el clasificador sea incapaz de discernir entre ellos. Por eso es muy importante el buscar atributos que puedan ser representativos de cada clase y permitan distinguirlas en la mayor medida posible.

Existen, además, dos problemas que pueden dificultar la obtención de clasificaciones correctas. El primero es la existencia de los llamados *outliers*, que son instancias cuyos valores se salen de lo esperado para la clase a la que pertenecen. También podemos

encontrarnos con el caso en el que, para una instancia dada, existan atributos con valor no definido, siendo entonces el clasificador el que decidirá qué hacer con ese ejemplo. Para cualquiera de estos dos casos puede optarse por filtrar esas muestras, lo que mejoraría los porcentajes de acierto para sistemas *offline*[5], pero no soluciona el problema para una aplicación en tiempo real.

El último punto a tratar en este apartado es la distinción entre los problemas uniclase y los multiclase. En los primeros, el objetivo es tratar de identificar si las muestras dadas pertenecen o no a una clase. Los problemas de clasificación multiclase tienen que tratar de dilucidar a qué clase pertenecen las instancias, de entre varias posibles. En lo que respecta a este proyecto, la mayoría de las clasificaciones realizadas son multiclase, ya que, se trata de identificar las vocales, las sílabas, u otros grupos que se verán en el capítulo 5 y en el anexo F. Sin embargo, también se han realizado pruebas que permitan distinguir entre muestras de pronunciación de sílabas contra señales que pertenecen a momentos en los que el sujeto no pronunciaba nada, éste es un caso de clasificación uniclase.

En el anexo E se puede encontrar más información sobre todo lo referente a clasificación y máquinas de aprendizaje comentado en este apartado.

4.4.2. Clasificadores

En este apartado van a explicarse los clasificadores con los que se han realizado los tests que se mostrarán en el capítulo 5. Consta de tres apartados, uno dedicado a cada método. Dada la naturaleza multiclase del problema ya presentado, el clasificador más sencillo que se puede utilizar es un árbol de decisión, por eso el primero en estudiar es uno de este tipo (apartado 4.4.2.1). Los clasificadores Bayesianos son también ampliamente utilizados en problemas de *machine learning* [22], de ellos el Naive Bayes es uno de los más utilizados (apartado 4.4.2.2). Por último, los algoritmos de *Boosting*, proveen métodos que tratan de mejorar el rendimiento de un clasificador dado, estudiaremos AdaBoost en el apartado 4.4.2.3.

4.4.2.1. Árbol de decisión J4.8

Un árbol de decisión es un clasificador no lineal, que nos provee una solución a un problema multiclase sirviéndose para ello de la filosofía *divide-and-conquer*[6]. Normalmente, en cada nodo se realiza la inspección de un atributo, comparándolo con un valor constante. Los nodos hoja son los que dan el resultado de la clasificación, aplicándose a todas las instancias que, siguiendo alguna rama del árbol, alcancen esa hoja. Por tanto, cada nuevo ejemplo a clasificar irá siguiendo una ruta, según los valores de sus atributos, y la hoja en la que finalmente termine determinará la clase a la que corresponde.

[5]Sistemas en los que el procesado y análisis de datos no se realiza en tiempo real.
[6]Divide y vencerás.

Tras la construcción del árbol, el siguiente paso es la generación de reglas, que serán las que realicen la clasificación. Cada uno de los nodos, podrá transformarse en una regla del estilo *if-then* que determinará el camino a seguir. Este es un procedimiento lento, lo que hace que la utilización de estos clasificadores sea más costosa en tiempo que, por ejemplo, el que veremos en el apartado 4.4.2.2.

Uno de los árboles de decisión más utilizados es el llamado C4.5, además de su sucesor comercial, el C5.0[23]. Ambos fueron creados por J. Ross Quinlan, quien ha trabajado en ellos durante unos 20 años. En su libro [24], se explica en profundidad el primero de ellos, donde incluye además el código fuente. El segundo está disponible comercialmente, los tests han detectado que no posee gran mejora en resultados de clasificación, sin embargo, la generación de reglas es notablemente más rápida.

La implementación en Weka del árbol de decisión C4.5 se llama J4.8. En realidad se trata de una versión posterior y mejorada, llamada C4.5 revisión 8, que es la última versión de esta familia de árboles antes del lanzamiento de la implementación comercial C5.0.

4.4.2.2. Naive Bayes

El clasificador Naive Bayes es un sencillo clasificador que se basa en la aplicación del teorema de Bayes, utilizando una simplificación *ingenua* (Naive). Se basa en asumir que los valores de los atributos $(a_1, a_2...a_n)$ que forman una muestra son independientes entre sí. Esto quiere decir que dada una instancia y una posible clase a la que puede pertenecer, la probabilidad de que esté formada por la conjunción $a_1, a_2...a_n$ es el producto de todas las probabilidades individuales que posee cada atributo de pertenecer a esa clase: $P(a_1, a_2...a_n|v_j) = \prod_i P(a_i|v_j)$.

El clasificador calculará la probabilidad de que una nueva muestra pertenezca a cada una de las clases existentes en el *universo* del problema y se tomará como resultado final la que ofrezca un valor más elevado. La fórmula del clasificador Naive Bayes es la siguiente:

$$v_{NB} = \text{máx}\, P(v_j) \prod_i P(a_i|v_j),$$

donde v_{NB} corresponde a la clase que Naive Bayes toma como salida, al dar la probabilidad más alta. Cada uno de los v_j son las clases que componen el problema, mientras que los a_i son los atributos que forman cada una de las muestras. El desarrollo de esta fórmula, junto con más información sobre los clasificadores Bayesianos puede consultarse en el anexo E.6.2.

4.4.2.3. AdaBoost

Boosting se refiere a los métodos que tratan de mejorar el rendimiento de un algoritmo de aprendizaje dado ([25], [26]). Se ha demostrado matemáticamente que a partir de un algoritmo de clasificación *débil* (weak learning algorithm), puede ser combinado para construir un clasificador *fuerte*, que mejore sus resultados. La idea es combinar una serie de hipótesis débiles ($h_1, h_2...h_n$) para que formen una hipótesis fuerte (h_F) cuyo rendimiento sea mejor que el de cada una de las débiles por separado.

$$h_F(x) = \sum_{i=1}^{n} w_i h_i(x),$$

donde w_i denota el peso asignado a la hipótesis h_i, siendo esto calculado por el algoritmo de Boosting. El resultado, h_F, es una suma ponderada de las hipótesis individuales.

AdaBoost[7] es uno de los algoritmos de Boosting más populares. Como entrada toma un conjunto de ejemplos de entrenamiento $(x_1, v_1)...(x_n, v_n)$, donde los x_i toman valores en un espacio X y v_i denota la clase a la que pertenece del universo V. Después se ejecutan una serie de iteraciones en las que para cada una se llama a un algoritmo débil para obtener una hipótesis. Inicialmente todas las hipótesis poseen el mismo peso, pero conforme se avanza en el entrenamiento adapta esos pesos según su rendimiento.

Para este proyecto se han probado el árbol de decisión J4.8 y Naive Bayes como clasificadores débiles y un número de iteraciones de 10, 50 y 100.

4.4.3. Weka

Weka es un software desarrollado por la Universidad de Waikato, en Nueva Zelanda, y su nombre proviene de *Waikato Environment for Knowledge Analysis*. Proporciona implementaciones de distintos algoritmos de clasificación que pueden ser fácilmente aplicadas a cualquier conjunto de datos. También incluye herramientas para transformar los datos, visualizarlos y analizar los resultados de los clasificadores. Puede verse en la figura 4.7 su pantalla principal.

Es una plataforma libre con Licencia General Pública GNU y está desarrollada en el lenguaje Java, lo que hace que sea un sistema muy portable entre distintos entornos. El problema que tiene es que, al estar desarrollada en Java, es bastante más lento de lo que sería una versión implementada en C o algún otro lenguaje de más bajo nivel. Es por esto que Weka no es una buena opción para un sistema en tiempo real, pero en cambio es un software ideal para estudios como el presente proyecto, en el que se requiere hacer pruebas distintas con varios clasificadores sobre muchos ficheros de datos. El tener todas

[7] Adaptative Boosting

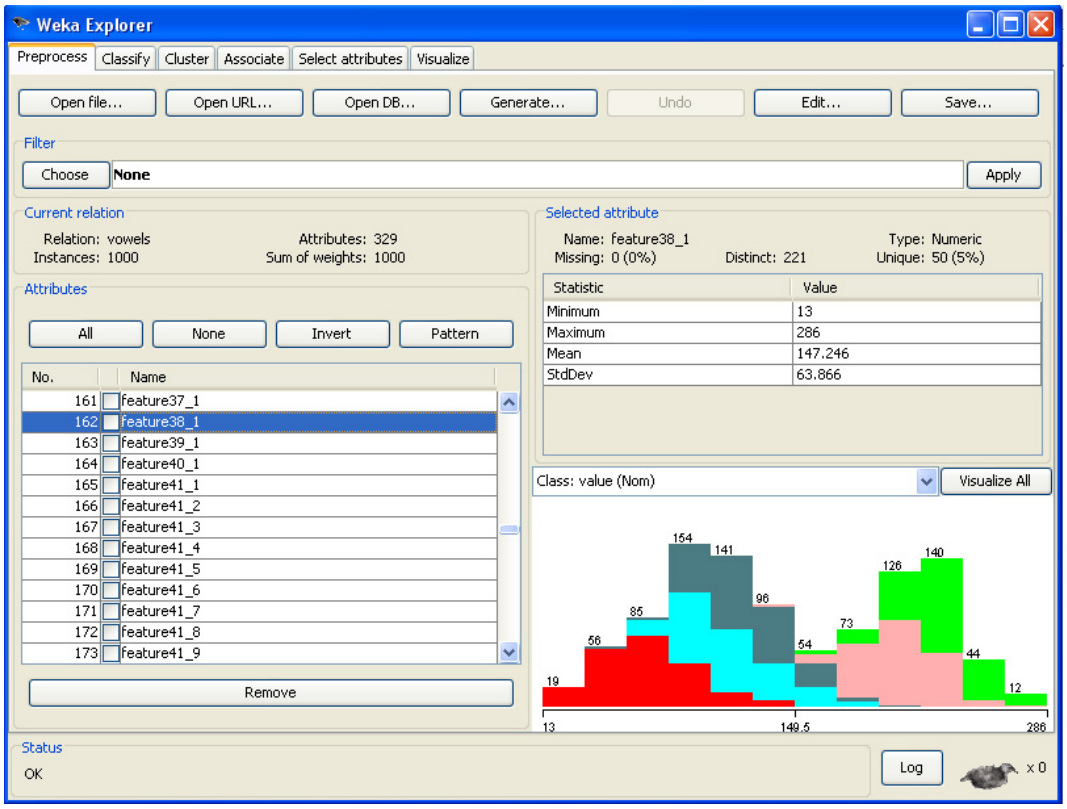


Figura 4.7: Pantalla principal de Weka. En la parte superior se encuentran las distintas pestañas y botones con opciones, por el centro aparecen las opciones de filtrado e información sobre los datos, en la parte izquierda la lista de características existentes y a la derecha se puede ver un histograma a color con los valores de la característica que se encuentre seleccionada.

las herramientas encapsuladas dentro de un mismo programa permite, por ejemplo, elegir cuál es el mejor clasificador sin tener que implementar cada uno de los que se desea probar.

Como ya se explicó en el apartado 4.4.2, los clasificadores que se han utilizado han sido el árbol de decisión J4.8, Naive Bayes y AdaBoost. Para todas las pruebas realizadas se ha empleado *10 fold cross-validation* (definición en E.5), ya que los resultados que proporciona son más realistas que los que se obtendría con una sola división aleatoria de los datos en un grupo para entrenar y otro para clasificar. Para cada ejecución Weka construye un modelo que muestra y en el que se puede ver, por ejemplo, cómo queda el árbol de decisión que implementa en el caso que se utilice el J4.8 y, con ello, cuáles eran las características que diferencian en mayor medida las distintas clases, ya que serán las que se encuentren en los nodos superiores.

5. Evaluación de los resultados

En este capítulo se van a presentar los resultados de los distintos clasificadores diseñados para resolver el problema presentado. En la sección 5.1 se explicarán los 3 clasificadores diseñados para resolver el problema de distinguir entre las 30 sílabas y los resultados proporcionados por cada uno. En la sección 5.2 se explicarán brevemente todas las pruebas realizadas antes de diseñar los clasificadores finales.

Todos los resultados de clasificación mostrados en este capítulo han sido obtenidos entrenando y clasificando cada uno de los esquemas propuestos con las 150 muestras de cada sílaba, empleando, en todos los casos, *10-fold cross validation*; lo que nos garantiza que los resultados sean más exactos por las 10 validaciones que realiza sobre estos. Además, todos los porcentajes de acierto mostrados en esta sección corresponden a la utilización de AdaBoost combinado con el árbol J4.8 aplicado en 100 iteraciones, ya que es la máquina que mejores porcentajes de acierto proporcionaba. La justificación y el resto de resultados pueden encontrarse en el anexo F.

5.1. Clasificación de sílabas

Como ya se ha indicado previamente en este texto, el problema principal que se desea resolver es tratar de distinguir muestras correspondientes a 30 sílabas o clases distintas. Para ello se han ideado 3 formas de realizar las clasificaciones. La primera y más sencilla (apartado 5.1.1) consiste en entrenar al clasificador con las 30 clases. Dado que 30 es un número muy elevado para un problema de este tipo, se decidió dividir el problema y diseñar otro que actúe de manera matricial (apartado 5.1.2), esto es, por un lado entrenar un clasificador que trate de reconocer las sílabas por su terminación y por el otro un clasificador que trate de reconocerlas por su comienzo, buscando la intersección de ambos se obtendría el resultado. Por último y dado que este clasificador, como se verá en el apartado 5.1.3, tiene ciertas limitaciones, se propusieron dos clasificadores que actúan con probabilidades condicionales.

5.1.1. Clasificador de 30 Clases

Este esquema de clasificación es el más sencillo e intuitivo que se puede diseñar (figura 5.1). Es un clasificador multiclase estándar y consiste en entrenar la máquina de aprendizaje con ejemplos de todas las clases existentes, en este caso 30. Su principal ventaja deriva de su sencillez, ya que una vez se ha completado el entrenamiento, la clasificación es directa al tener solamente un clasificador. Como inconveniente se puede remarcar que para un sistema con un número de clases muy elevado puede resultar mucho más lento al tener que realizar todo el trabajo una sola máquina.

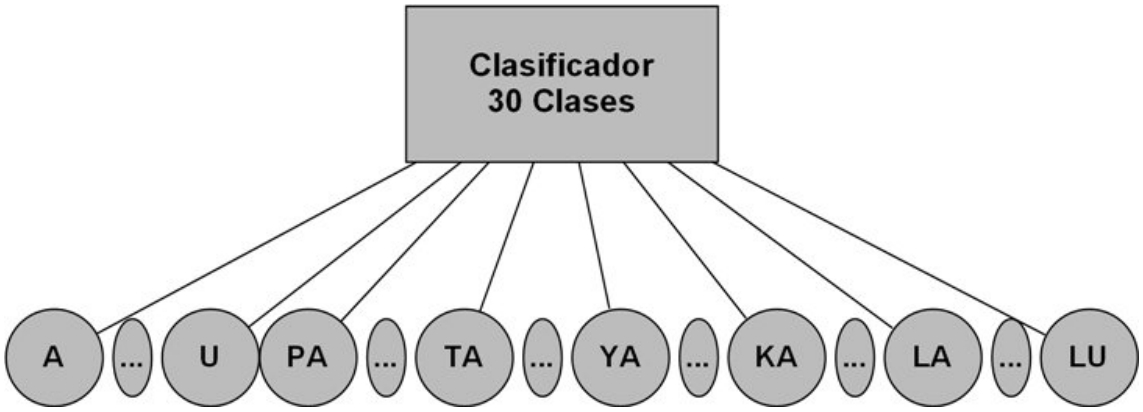


Figura 5.1: Esquema correspondiente al clasificador multiclase estándar de 30 clases.

Para obtener los resultados se ha utilizado el clasificador con los 150 ejemplos de cada una de las 30 clases, lo que hace un total de 4.500 muestras. Al usar *10-fold cross validation* cada una de las muestras se utiliza 9 veces para entrenar y otra para testear, así se consigue que el porcentaje final calculado sea más representativo. En la figura 5.2 se muestra, mediante una gráfica a color, la matriz de confusión que se ha obtenido con este esquema de clasificación, donde los tonos rojizos representan un porcentaje de acierto más próximo al 100 %. El porcentaje medio de acierto es de un **70,93 %**. El valor máximo de *true positives* se consigue en la clase *A*: 95,33 %, el mínimo corresponde a la clase *TI*: 46 %.

La matriz está estructurada ordenando las sílabas por terminación, así pueden observarse las agrupaciones que se producen en los resultados de clasificación. Como puede comprobarse, la mayoría de los errores se producen entre sílabas con la misma terminación, por eso se forman esos cuadrados (*TA*, *YA*, *KA*, *LA*; *TE*, *YE*, *KE*, *LE*; etc). También se puede concluir que las vocales y las sílabas que comienzan por *P* se distinguen con mucha más facilidad que el resto, algo que, seguramente, está causado porque son los dos únicos grupos en los que no se emplea la lengua para su gesticulación. El resto de grupos se confundirán más fácilmente, ya que gran parte de la información que produce su pronunciación (piénsese en *TA*, *YA*, *KA*, *LA*...) es llevada a cabo por la lengua, lugar bastante difícil de medir y para el que únicamente se dispone de un canal situado bajo la barbilla. Las líneas paralelas a la diagonal principal que aparecen son debidas a las confusiones que

se producen por terminaciones similares; las más significativas son las que aparecen entre las terminaciones *E-I*, aunque también surgen, en menor medida entre *O* y *U* y alguna entre *A* y *E*.

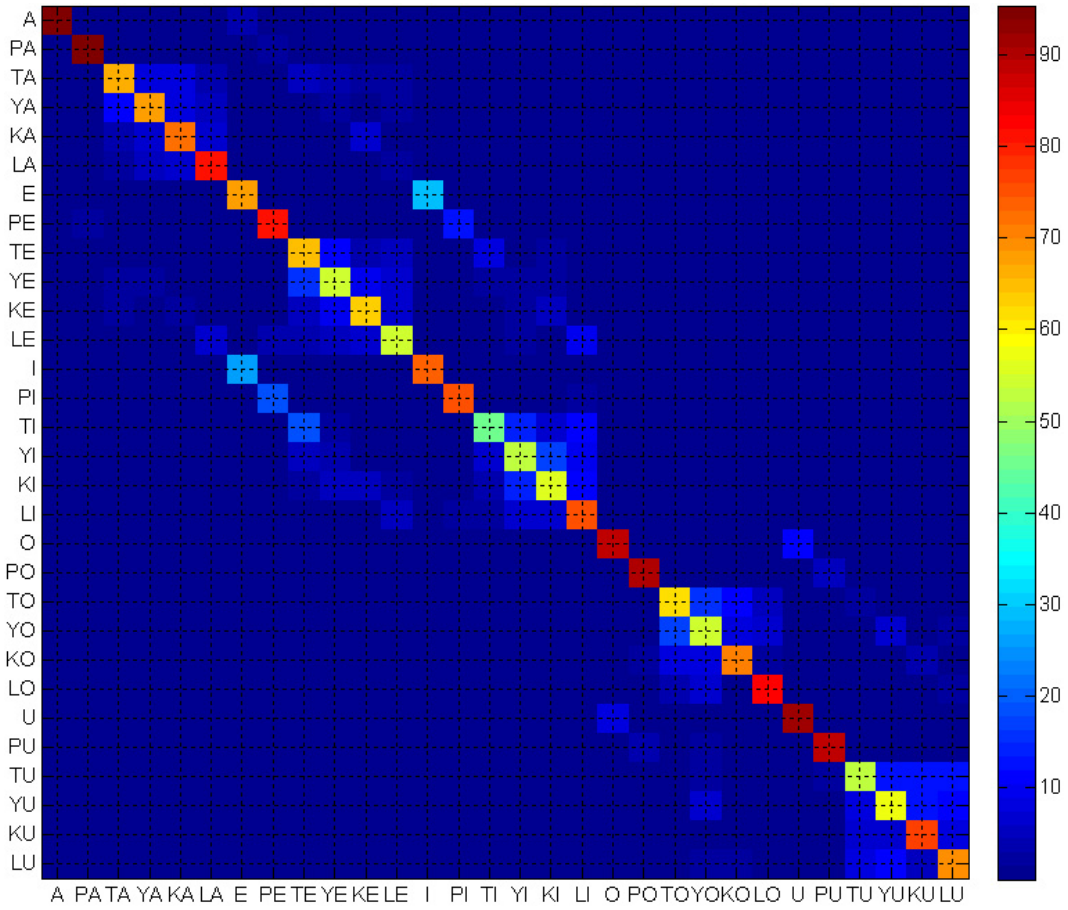


Figura 5.2: Matriz de confusión correspondiente al clasificador de 30 clases.

5.1.2. Clasificador Matricial

El esquema de clasificación matricial consiste en una división simple del problema en dos más pequeños. En la matriz de confusión mostrada en la figura 5.2 se observaba que prácticamente todos los errores se producen al confundirse entre clases que terminan o que empiezan igual. Esto motiva la idea de crear dos clasificadores multiclase distintos, uno especializado en distinguir los comienzos y el otro en las terminaciones (figura 5.3). Por tanto, el primer clasificador, al que también llamaremos clasificador por filas, tratará de distinguir entre 6 clases: vocales, sílabas que empiezan por *P*, *T*, *Y*, *K* y *L*. El segundo, clasificador por columnas, deberá diferenciar las sílabas por su terminación, así que tendrá 5 clases según terminen en *A*, *E*, *I*, *O* o *U*.

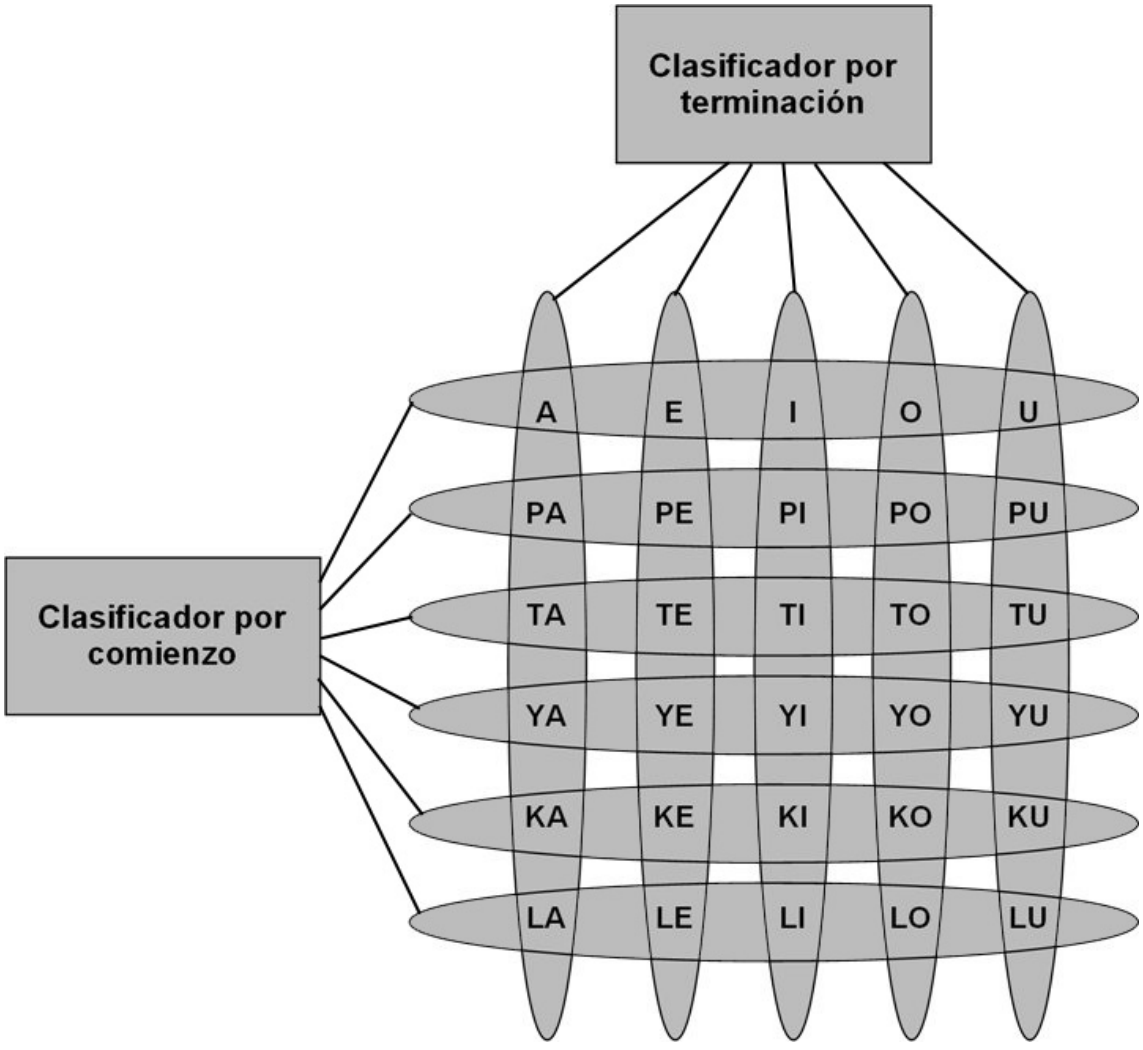


Figura 5.3: Esquema correspondiente al clasificador matricial formado por la intersección de dos clasificadores multiclase: uno por filas y otro por columnas.

Para el clasificador por filas se utilizarán 750 ejemplos para cada una de sus 6 clases, ya que éstas son uniones de grupos de 5 sílabas cada una. El clasificador por columnas

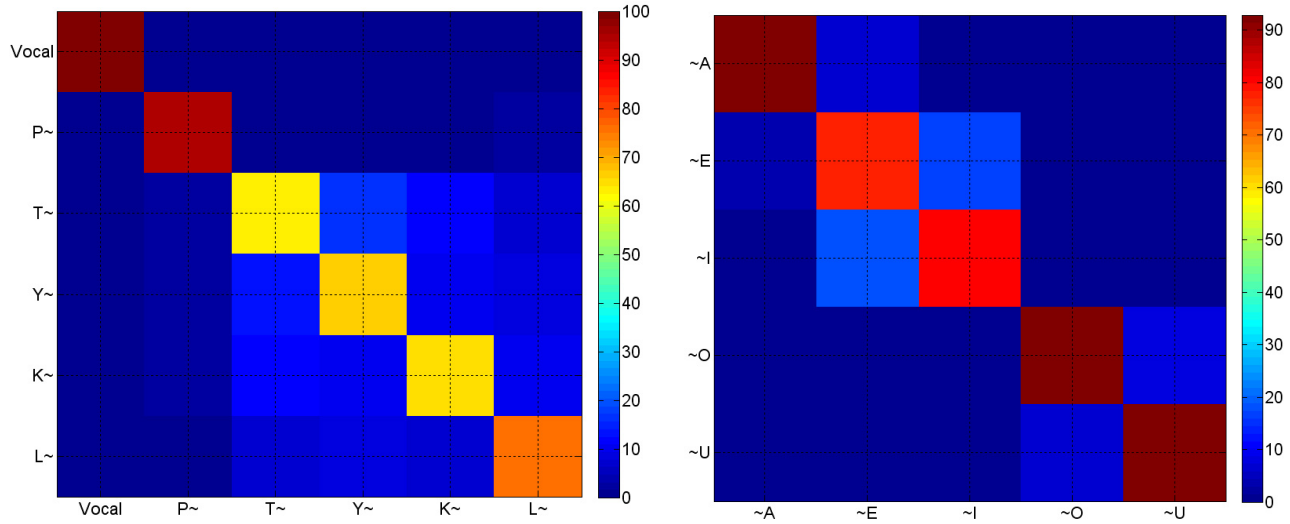
empleará para cada una de las 5 clases que posee 900 ejemplos, debido a que cada clase está formada por 6 sílabas.

La imagen 5.4 muestra las matrices de confusión correspondientes a los dos clasificadores. El que actúa por filas, consigue acertar en el 77,47 % de los casos, mientras que el que trabaja por columnas lo hace el 87,33 %. En este esquema suponemos que la probabilidad de que una sílaba tenga un comienzo y una terminación determinada es el producto de las probabilidades de que suceda cada cosa por separado:

$$P(c,t) = P(c)P(t),$$

por tanto, multiplicando ambos porcentajes se obtiene un acierto del clasificador matricial, que es del **67,65 %**.

Pese a la división del problema en dos, nos encontramos con las mismas confusiones que en el esquema previo. Con respecto al clasificador por comienzos, vemos que se forma un cuadrado de forma análoga a como se formaba con el clasificador de 30 clases, agrupándose por un lado las sílabas que empiezan por *T*, *Y*, *K* y *L* y distinguiéndose con claridad las vocales y las que empiezan por *P*. Por otro lado, el clasificador de terminaciones también confunde las sílabas que acaban en *E*-*I* y, en menor medida, las que finalizan en *O*-*U* y *A*-*E*.



(a) (b)

Figura 5.4: Matrices de confusión correspondientes a los clasificadores que forman el esquema matricial. La matriz (a) pertenece al clasificador por comienzos y la matriz (b) al clasificador por terminaciones.

Para una implementación del esquema en tiempo real sería indiferente el orden en el que se lanzasen los clasificadores, ya que el cálculo de la probabilidad final no varía por ser simplemente la multiplicación de los dos resultados individuales. De ahí deriva el principal problema de este esquema, que se entenderá mejor con el siguiente ejemplo: pongamos

que queremos reconocer una sílaba *TA*, lanzamos primero el clasificador de comienzos y determina que empieza por *T*; lo siguiente es lanzar el clasificador de terminaciones, pero éste ha sido entrenado con todas las sílabas que terminan en cada vocal y no se está aprovechando la información disponible correspondiente al resultado obtenido por el primer clasificador, es decir, para incrementar más la probabilidad de acierto, se debería emplear en segundo lugar un clasificador condicionado al resultado del primero y que, en este ejemplo, sólo trataría de distinguir entre *TA*, *TE*, *TI*, *TO* y *TU*.

5.1.3. Clasificadores Condicionales

Como solución al problema encontrado en el esquema del apartado anterior nace la idea de implementar clasificadores condicionales. El funcionamiento básico es muy similar al matricial, pero el número de clasificadores empleados varía dependiendo de cómo se plantee el problema. Se dispone de dos opciones, según cuál se lance primero.

Si se ejecuta primero el clasificador por filas (punto 5.1.3.1), éste proporcionará como resultado cuál cree que es el comienzo de la sílaba, por tanto, después hay que tener tantos clasificadores condicionales como posibles comienzos, en este caso 6 (Vocal, P, T, Y, K, L). Cada uno de ellos estará entrenado solamente con muestras de sílabas que poseen el comienzo que ha determinado el primer clasificador.

En el caso de que se lance primero la máquina encargada de reconocer las terminaciones (punto 5.1.3.2), se necesitarán después 5 clasificadores condicionales (∼A, ∼E, ∼I, ∼O, ∼U), cada uno de ellos entrenado con sólo sílabas pertenecientes a esas terminaciones.

Supuestamente esto debería mejorar los resultados del clasificador matricial, ya que se descarta información inútil al no entrenar los clasificadores condicionales con muestras que quedan descartadas por el primero de ellos.

5.1.3.1. Clasificador Fila-Columna Condicional

Para este esquema el clasificador de filas o comienzos es exactamente el mismo que se utiliza en el clasificador matricial del apartado 5.1.2, pero en vez de tener un solo clasificador por columnas se utilizan 6, uno por cada posible comienzo (figura 5.5). Así, habrá un clasificador entrenado específicamente para reconocer vocales, otro para sílabas que empiecen por *P*, *T*, etc. Por tanto, el cálculo de la probabilidad de que una sílaba dada posea un comienzo y un final determinado es

$$P(c,t) = P(t|c)P(c),$$

es decir, la probabilidad condicionada de la terminación con respecto al comiendo multiplicada por la probabilidad de ese comienzo.

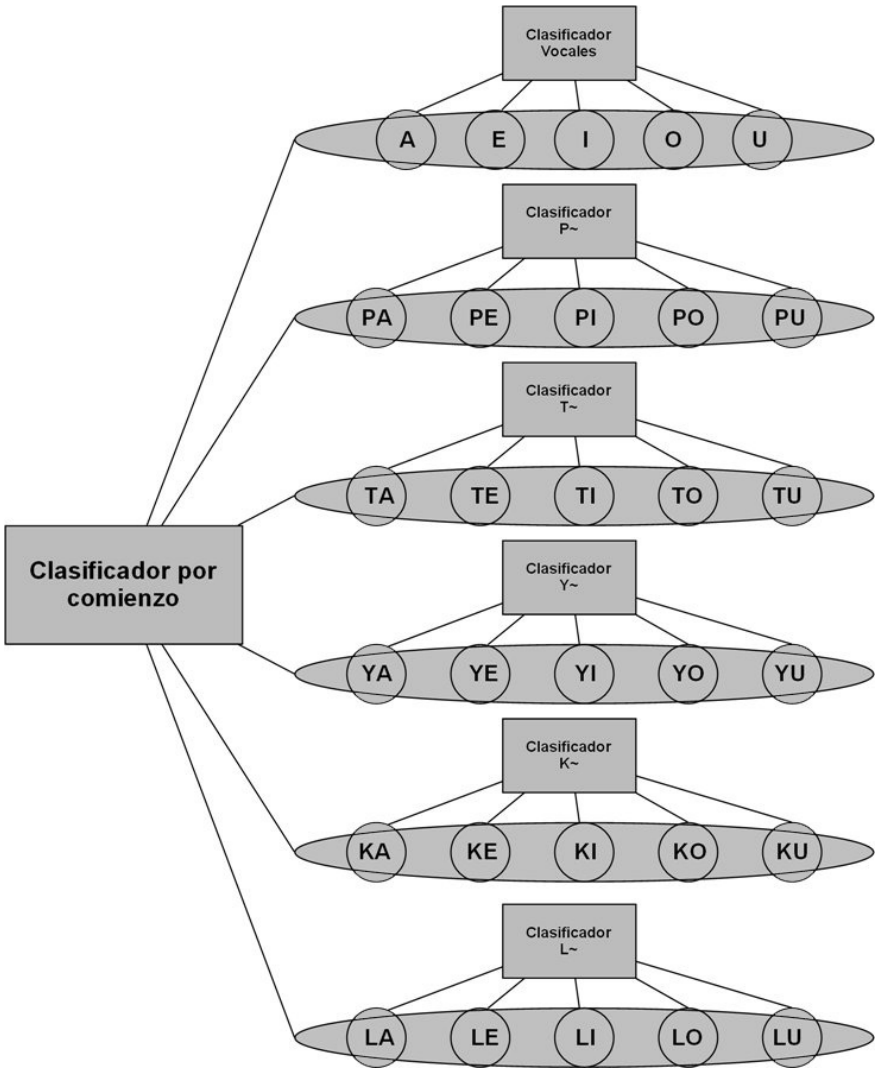


Figura 5.5: Esquema correspondiente al clasificador condicional compuesto por un clasificador por filas y 6 por columna, cada uno correspondiente a un comienzo distinto.

En la figura 5.7 se muestran en forma de gráfica los porcentajes de acierto obtenidos por cada uno de los clasificadores condicionales, que varían entre el 81,73 % y el 92 %, siendo la media de los 6 un 88,93 %. Por tanto, multiplicando este valor por la probabilidad que proporcionaba el clasificador de comienzos, que era del 77,47 %, obtenemos un resultado final del **68,89 %**. La matriz de confusión correspondiente al clasificador por columnas encargado de distinguir las terminaciones de las sílabas que comienzan por *P* puede observarse en la figura 5.6, las del resto de clasificadores columna condicionales aparecen en el anexo F.3.3.1. La del clasificador por filas se encuentra en la figura 5.4(a).

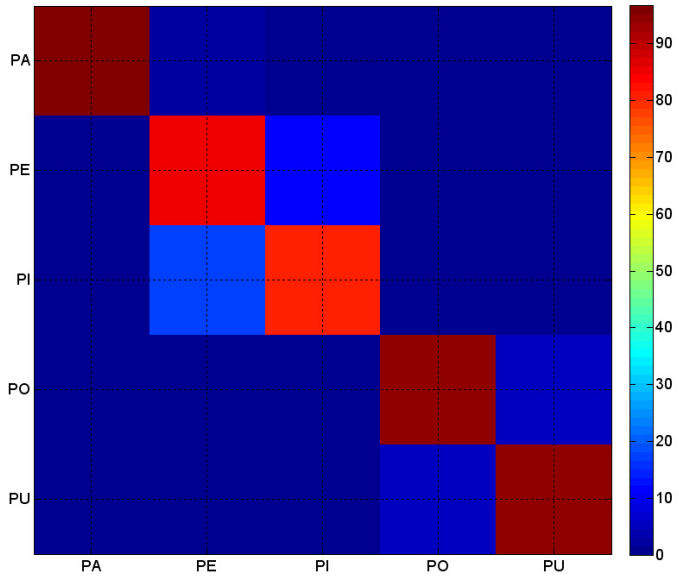


Figura 5.6: Matriz de confusión correspondiente al clasificador condicional de sílabas que comienzan por *P*.

Para una implementación real de este esquema, al igual que en el caso del siguiente, podría ejecutarse en varios procesadores en paralelo, ahorrándose tiempo en el caso de que se tuviese un número mucho mayor de clases. El problema es que para un número reducido de éstas, probablemente no compense tener que entrenar y seleccionar entre 6 clasificadores extra.

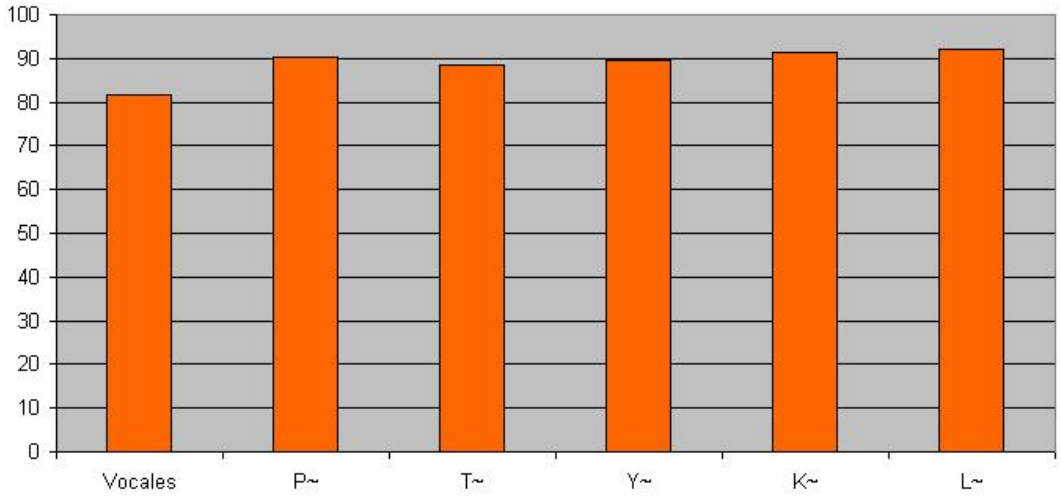


Figura 5.7: Diagrama de barras correspondiente a los porcentajes de *true positives* obtenidos por cada uno de los 6 clasificadores condicionales de columnas.

5.1.3.2. Clasificador Columna-Fila Condicional

El funcionamiento de este esquema de clasificación se basa en la misma idea que el anterior, la única diferencia es que en este caso se ejecutará en primer lugar el clasificador que distingue por columnas o terminaciones y después actuará, dependiendo de la respuesta del primero, una de las 5 máquinas entrenadas para distinguir los comienzos (figura 5.8). La razón por la que se han probado los dos esquemas es porque, a priori, no se podía determinar si uno iba a dar mejores resultados que el otro y, de ser así, cuál sería mejor.

En este caso, el cálculo de la probabilidad de acierto para una sílaba será la probabilidad condicionada de un comienzo con respecto a una terminación dada, multiplicado por la probabilidad de esa terminación:

$$P(c,t) = P(c|t)P(t)$$

La matriz de confusión correspondiente al clasificador entrenado para distinguir entre las sílabas que terminan por *A* está en la figura 5.9, las del resto de terminaciones pueden consultarse en el anexo F.3.3.2, mientras que la del clasificador columna aparece en la figura 5.4(b).

Los ratios de clasificación proporcionados por cada uno de los 5 clasificadores específicos pueden verse en la figura 5.10 y están entre el 76,44 y el 84,22 %, situándose la media en el 79,56 %. Si multiplicamos este valor por el porcentaje de acierto que ofrece el clasificador por columnas (87,33 %), obtenemos como resultado un acierto medio del **69,48 %**. Este resultado, pese a ser mayor que el obtenido por el esquema previo, la diferencia no es tan significativamente grande como para concluir que es mejor.

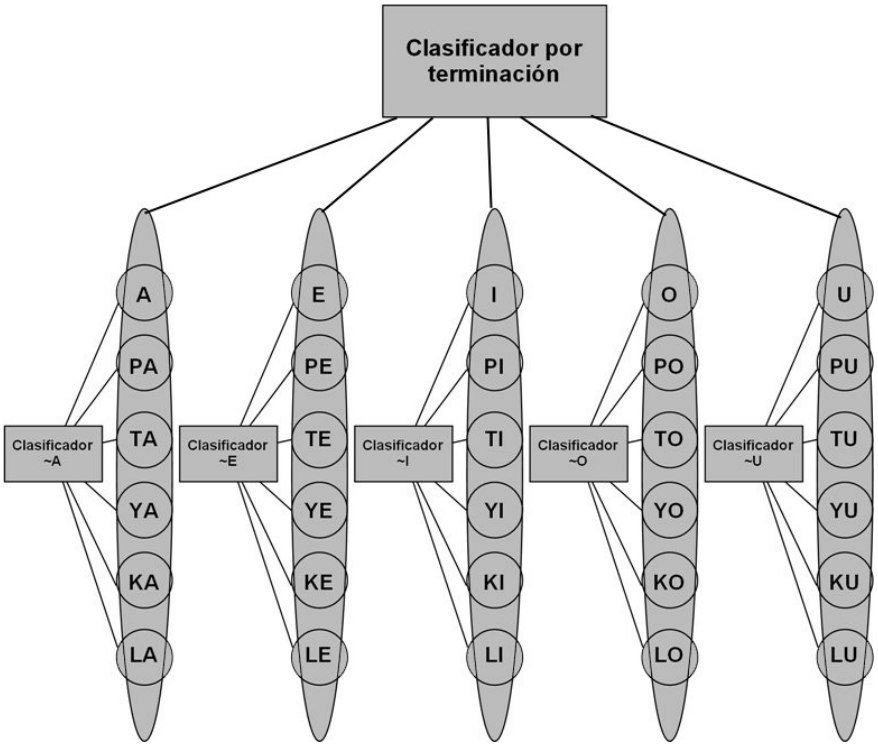


Figura 5.8: Esquema correspondiente al clasificador condicional compuesto por un clasificador por columnas y 5 por fila, cada uno correspondiente a una terminación diferente.

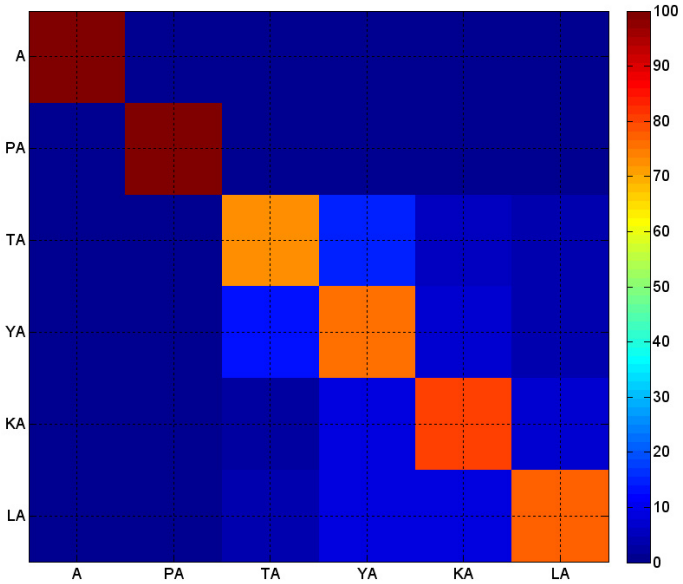


Figura 5.9: Matriz de confusión correspondiente al clasificador condicional de sílabas que terminan por *A*.

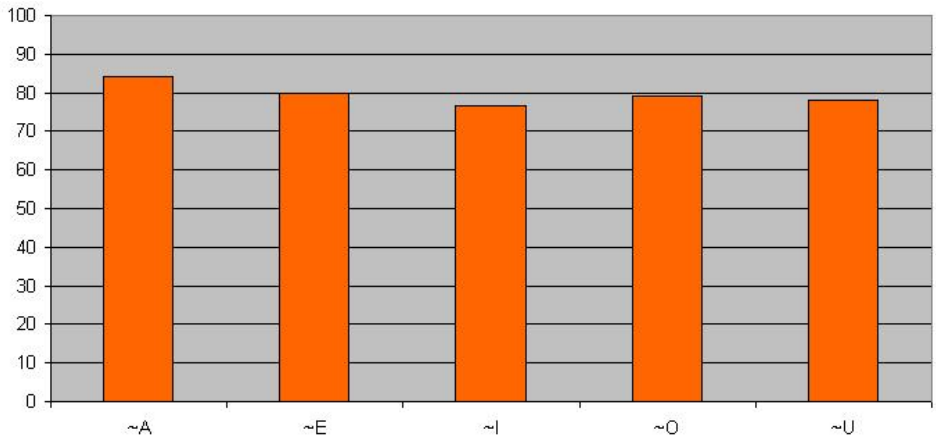


Figura 5.10: Diagrama de barras correspondiente a los porcentajes de *true positives* obtenidos por cada uno de los 5 clasificadores condicionales por fila.

5.1.4. Comparativa

Una vez explicados los tres esquemas diseñados y mostrados los resultados obtenidos para cada uno, se va a realizar una comparativa entre ellos. En la figura 5.11 se muestra un diagrama de barras en el que se ven los porcentajes de acierto obtenidos con cada método. Lo primero que se observa es que la diferencia entre el que proporciona el mejor (70,93 %) y el peor resultado (67,65 %) es menor que un 3,5 %, con lo que se puede concluir que éste no sería un criterio definitivo para seleccionar uno u otro esquema.

Otro punto en el que pueden compararse es el tiempo de entrenamiento de cada uno de los esquemas de clasificación. Supuestamente, la máquina de aprendizaje sólo tiene que entrenarse una vez y después está lista para clasificar infinidad de ejemplos, sin embargo, para máquinas poco potentes este tiempo puede llegar a ser excesivamente largo. Para hacerse una idea de la diferencia entre unos y otros, con un procesador a 2,50GHz, utilizando el método de clasificación AdaBoost + J4.8 + 100 iteraciones, cada clasificador condicional tardaría en ser entrenado unos 4 o 5 minutos, uno matricial tardaría del orden de 30 minutos y el de 30 clases unas 2 horas. La ventaja que tienen el esquema matricial y, sobre todo, los condicionales, es que cada clasificador podría entrenarse y ejecutarse en un procesador distinto, lo que mejoraría las prestaciones temporales en gran medida.

Llegados a este punto, la decisión entre uno u otro esquema depende mucho de cómo vaya a expandirse este proyecto. Pero lo más natural sería que conforme se incremente considerablemente el número de clases, el clasificador multiclase estándar explicado en el apartado 5.1.1 reduzca sus prestaciones en un grado mucho mayor de lo que lo hagan el matricial o uno condicional, por tanto, seguramente, ése sería el primero en desechar. Para decantarse por uno de los restantes habría que estudiar en profundidad el equipo del que se dispone, ya que para un dispositivo muy básico lo más probable es que los condicionales sobrecarguen demasiado el sistema y el matricial proporcionaría un equilibrio bastante estable pese a que, precisamente, éste es el esquema con el que se han obtenido unos ratios

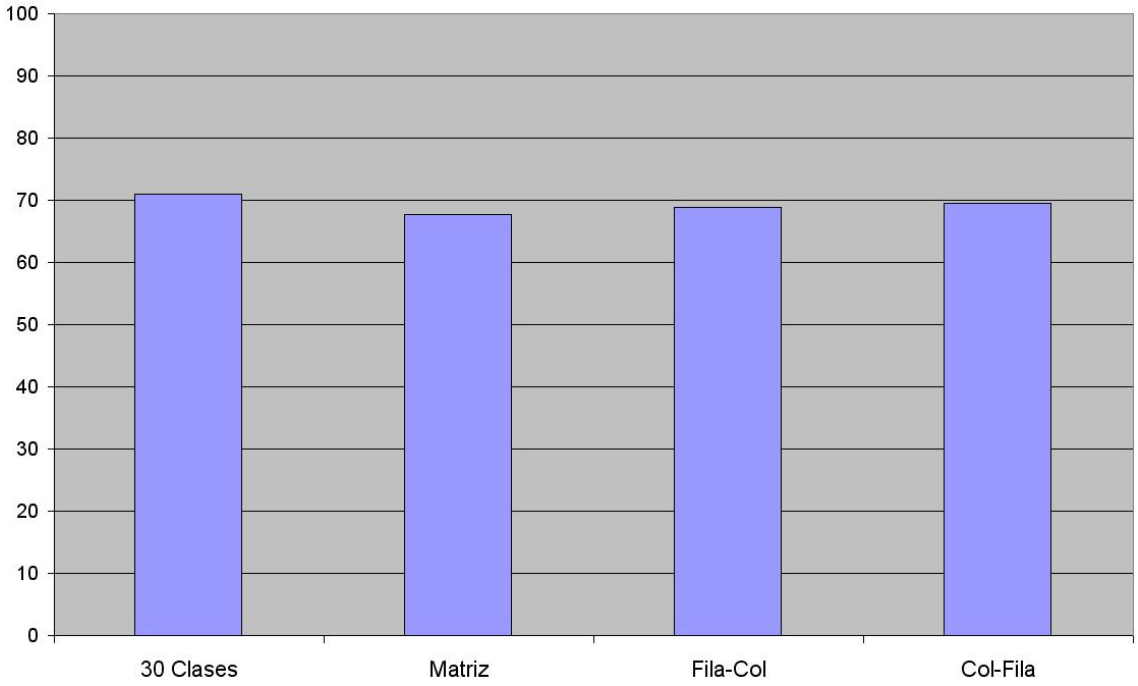


Figura 5.11: Comparativa de los resultados de acierto obtenidos por cada uno de los 4 esquemas de clasificación estudiados.

de acierto menores.

5.2. Resultados adicionales

Antes de diseñar un clasificador que trate de diferenciar entre las 30 sílabas que componen nuestro universo de clases, se han realizado distintas pruebas para resolver problemas más pequeños. Así, paso a paso, se ha ido buscando la mejor metodología con la que abordar el problema de clasificación. Para esto se realizaron tests con sólo las 5 vocales, donde el mejor resultado obtenido fue el acierto en el 80,2 % de los casos.

Además se han diseñado clasificadores para distinguir pronunciaciones de vocal contra sílaba consonántica y para diferenciar señales pertenecientes a una pronunciación de las que no lo son, obteniendo para ambos tests resultados superiores al 99 %.

Pese a que es una tarea fuera del alcance de este proyecto, se han realizado pruebas para conseguir una primera aproximación sobre si influye en los rendimientos la mezcla de datos adquiridos en distintos días. Éstas arrojaron unos resultados muy optimistas, demostrando que, pese a las pequeñas variaciones en la posición de los sensores que pueden producirse, los resultados en las clasificaciones no varían significativamente. Puede encontrarse más información y los resultados exactos de estos tests en la sección 2 del anexo F.

6. Conclusiones y trabajo futuro

La principal conclusión que puede extraerse de la evaluación del proyecto es que se han cumplido los objetivos propuestos: se ha diseñado un prototipo capaz de reconocer un conjunto de 30 sílabas simples y se ha validado, obtenido un rendimiento de alrededor del 70 % de acierto. Teniendo en cuenta la complejidad que tiene un problema de clasificación con tantas clases, el porcentaje conseguido es realmente satisfactorio.

Además, debe remarcarse especialmente el hecho de que, pese a que un 70 % puede parecer un resultado no muy alto, la mayoría de los errores que se producen en los clasificadores son razonablemente aceptables, ya que las confusiones que más se repiten son entre sílabas que, si bien su pronunciación sonora es ya parecida, más todavía lo es su gesticulación. Piénsese, por ejemplo, en las sílabas *TI* y *YI*, que son dos de las que más se confunden mutuamente. Por tanto, en versiones futuras de la prótesis, fallos de este tipo podrían ser solventados al conocer cuáles son las sílabas que más se confunden entre sí y depender el reconocimiento del contexto en el que se encuentra cada una al concatenarlas para formar palabras.

Dado que no hemos encontrado constancia de ningún proyecto similar desarrollado en castellano, la comparativa de los resultados obtenidos con otros sistemas no es directa. Sin embargo, pueden compararse con prototipos realizados en distintos idiomas, pese a que el vocabulario que reconocen sea bastante distinto al expuesto aquí. Los resultados obtenidos en este proyecto son coherentes con los que se han observado en los trabajos referenciados durante el texto. Con respecto a los sistemas de reconocimiento de vocales, nos encontramos con unos resultados similares a los que se consiguen en otros idiomas. El mejor de los estudiados ([11]), con un 90 %, lo consigue gracias a un filtrado manual de las señales defectuosas, cosa que se descartó realizar en este proyecto dado que, a la hora de hacer la implementación en tiempo real, es imposible hacer ese filtrado y, por tanto, los resultados ya no serían los mismos. En sistemas de clasificación de palabras los resultados dependen mucho tanto del número de palabras que compongan el vocabulario como de lo diferentes que sean las palabras a reconocer. Por mencionar tres sistemas: en [6], con un vocabulario de 6 palabras, clasifican al 90 %; [8], clasifican 15 palabras acertando el 74 % de las veces; y [13], reconocen 18 palabras con un rendimiento del 77.8 % .

Como ya se indicó en el capítulo 2, el desarrollo completo de una prótesis del habla mediante electromiografía es tremendamente ambicioso y requerirá de varios proyectos

más hasta conseguir una versión completamente operativa, sin embargo, este trabajo pretende ser la base en la que apoyarse para futuras investigaciones y expansiones.

Bibliografía

[1] F. Gimeno Perez and B. Torres Gallardo. *Anatomía de la voz*. Paidotribo, 1 edition, September 2008.

[2] S. Rodriguez and J. M.ª Smith-Agreda. *Anatomía De Los Órganos Del Lenguaje, Visión Y Audición*. Editorial Médica Panamericana S.A., 2 edition, 2004.

[3] Antonio Ríos Mestre. *La transcripción fonética automática del diccionario electrónico de formas simples flexivas del español: estudio fonológico en el léxico*. Laboratorio de Lingüística Informática de la Universidad Autónoma de Barcelona, 4 edition, 1999.

[4] Sridhar P Arjunan, Hans Weghorn, Dinesh K. Kumar, and Wai C. Yau. Vowel recognition of English and German language using Facial movement(SEMG) for Speech control based HCI. *HCSNet Workshop on the Use of Vision in HCI*, 2006.

[5] Sridhar P. Arjunan, Dinesh K. Kumar, Wai C. Yau, and Hans Weghorn. Unspoken Vowel Recognition Using Facial Electromyogram. *Proc. of the 28th IEEE EMBS Annual International Conference*, September 2006.

[6] Chuck Jorgensen, Diana D. Lee, and Shane Agabon. Sub Auditory Speech Recognition Based on EMG/EPG Signals. *Proc. IJCNN*, January 2003.

[7] Sanjay Kumar, Dinesh K. Kumar, Melaku Alemu, and Mark Burry. EMG Based Voice Recognition. *Proc. ISSNIP Conference*, pages 593–598, 2004.

[8] Bradley J. Betts, Kim Binsted, and Charles Jorgensen. Small-vocabulary speech recognition using surface electromyography. *Interacting with Computers*, 18(6):1242–1259, December 2006.

[9] Ki-Seung Lee. EMG-Based Speech Recognition Using Hidden Markov Models With Global Control Variables. *IEEE Transactions on Biomedical Engineering*, 55(3), July 2007.

[10] Szu-Chen Stan Jou and Tanja Schultz. Automatic Speech Recognition based on Electromyographic Biosignals. *Biomedical Engineering Systems and Technologies International Joint Conference*, January 2008.

[11] José AG Mendes, Ricardo R. Robson, Sofiane Labidi, and Allan Kardec Barros. Subvocal Speech Recognition Based on EMG signal Using Independent Component Analysis and Neural Network MLP. *Congress on Image and Signal Processing*, pages 221–224, 2008.

[12] A. D. C. Chan, K. Englehart, B. Hudgins, and D. F. Lovely. A Multi-Expert Speech Recognition System using Acoustic and Myoelectric Signals. *Proceedings of the Second Joint EMBS/BMES Conference*, 2002.

[13] Quan Zhou, Ning Jiang, Kevin Englehart, and Bernard Hudgins. Improved Phoneme-Based Myoelectric Speech Recognition. *IEEE Transactions on Biomedical Engineering*, 2009.

[14] B.G. Lapatki, D.F. Stegeman, and I.E. Jonas. A surface EMG electrode for the simultaneous observation of multiple facial muscles. *Journal of Neuroscience Methods*, 123:117/128, October 2002.

[15] Cheng-Ning Huang, Chun-Han Chen, and Hung-Yuan Chung. The Review of Applications and Measurements in Facial Electromyography. *Journal of Medical and Biological Engineering*, 25(1):15–20, November 2004.

[16] Hermie J. Hermens, Bart Freriks, Catherine Disselhorst-Klug, and Günter Rau. Development of recommendations for SEMG sensors and sensor placement procedures. *Journal of Electromyography and Kinesiology*, 10:361–374, 2000.

[17] Travis W. Beck, Terry J. Housh, Joel T. Cramer, Moh H. Malek, Michelle Mielke, Russell Hendrix, and Joseph P. Weir. A comparison of monopolar and bipolar recording techniques for examining the patterns of responses for electromyographic amplitude and mean power frequency versus isometric torque for the vastus lateralis muscle. *Journal of Neuroscience Methods*, 166:159–167, 2007.

[18] Alan J. Fridlund and John T. Cacioppo. Guidelines for Human Electromyographic Research. *Psychophysiology*, 23(5):567–589, September 1986.

[19] G. Shalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, and J.R. Wolpaw. BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. *IEEE Transactions on Biomedical Engineering*, 51(6), May 2004.

[20] Guido Dornhege, José del R. Millán, Thilo Hinterberger, Dennis J. McFarland, and Klaus-Robert Müller. *Toward Brain-Computer Interfacing.* 2007.

[21] S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Academic Press, 3 edition, 2006.

[22] Tom M. Mitchell. *Machine Learning.* Mc Graw-Hill International Editions, 2003.

[23] Ian H. Witten and Eibe Franck. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Publishers, 2 edition, 2005.

[24] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, 1993.

[25] Óscar Martínez Mozos. Semantic Labeling of Places with Mobile Robots. July 2008.

[26] Robert D. Vincent, Joelle Pineau, Philip de Guzman, and Massimo Avoli. Recurrent Boosting for Classification of Natural and Synthetic Time-Series Data. *Lecture Notes on Artificial Intelligence*, pages 192–203, 2007.

[27] H. Manabe and Z. Zhang. Multi-stream HMM for EMG-Based Speech Recognition. *International Conference of the IEEE EMBS*, September 2004.

[28] Michael J. Kearns and Leslie G. Valiant. Learning boolean formulae or finite automata is as hard as factoring. Technical report, Harvard University Aiken Computation Laboratory, 1988.

[29] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, pages 197–227, 1990.

Anexo A. Desarrollo

En este anexo se va a detallar cómo se ha desarrollado cada uno de los pasos que han formado este proyecto final de carrera. La idea de comenzar a trabajar en este proyecto surgió en septiembre de 2008, siendo para la segunda semana del mes de noviembre cuando realmente comenzó el trabajo, que se ha extendido hasta agosto del año 2009.

A.1. Hitos del proyecto

A continuación se van a describir los principales hitos que han formado parte del desarrollo del presente trabajo. Pese a que se van a comentar las tareas principales organizadas por meses, hay que mencionar que algunas de ellas han sido realizadas de manera más o menos continua, aunque hayan tenido un esfuerzo mayor en determinados momentos.

En **noviembre de 2008** es donde se inicia el trabajo; al tratarse de un proyecto de investigación, los dos primeros meses se han dedicado exclusivamente a la lectura y el estudio sobre tecnologías EMG, reconocedores de voz, sistemas de procesamiento de señal, etc. Este paso es realmente imprescindible, ya que comenzar a trabajar sin haber plantado una buena base de conocimiento hubiese supuesto mayores pérdidas de tiempo a posteriori. Sin embargo, durante todo el transcurso del proyecto, el autor ha continuado buscando y leyendo información que pudiese resultar relevante para la mejora de algún aspecto del trabajo que se iba desarrollando.

Para comienzos de **enero de 2009** se inicia el diseño e implementación de las aplicaciones gráficas, escritas en Matlab, que se encargarán de realizar los tratamientos de señal. Este trabajo se fue compaginando con el estudio sobre máquinas de aprendizaje, que sería la tecnología a utilizar cuando acabase el desarrollo de las aplicaciones señaladas.

Es para el **23 de marzo** cuando se ha concluido la etapa anterior y se procede a diseñar un primer experimento de adquisición de datos. Se emplea un mes para el análisis de resultados de lo adquirido en esas sesiones hasta que se decide realizar una modificación en la metodología de adquisición de datos EMG de monopolar a bipolar.

Alrededor del **20 de abril** se comienza la adaptación del software de tratamiento de señales, para compatibilizarlo con las señales bipolares, tarea que dura un mes y que,

después, requirió de otra sesión sencilla de experimentación para verificar el correcto funcionamiento.

Los días **1** y **22 de junio** y el **6 de julio** se realizaron las sesiones de adquisición de datos. Las muestras obtenidas en la primera de ellas tuvieron que desecharse por problemas técnicos en la experimentación. La primera de las fechas marcó el inicio de la última fase en la que se comenzó la obtención y análisis de resultados de clasificación y que duraría alrededor de dos meses.

Por último, para el **20 de julio** comenzó el proceso de redacción de este documento, proceso que duraría hasta prácticamente finales de agosto de 2009.

A.2. Diagrama de Gantt

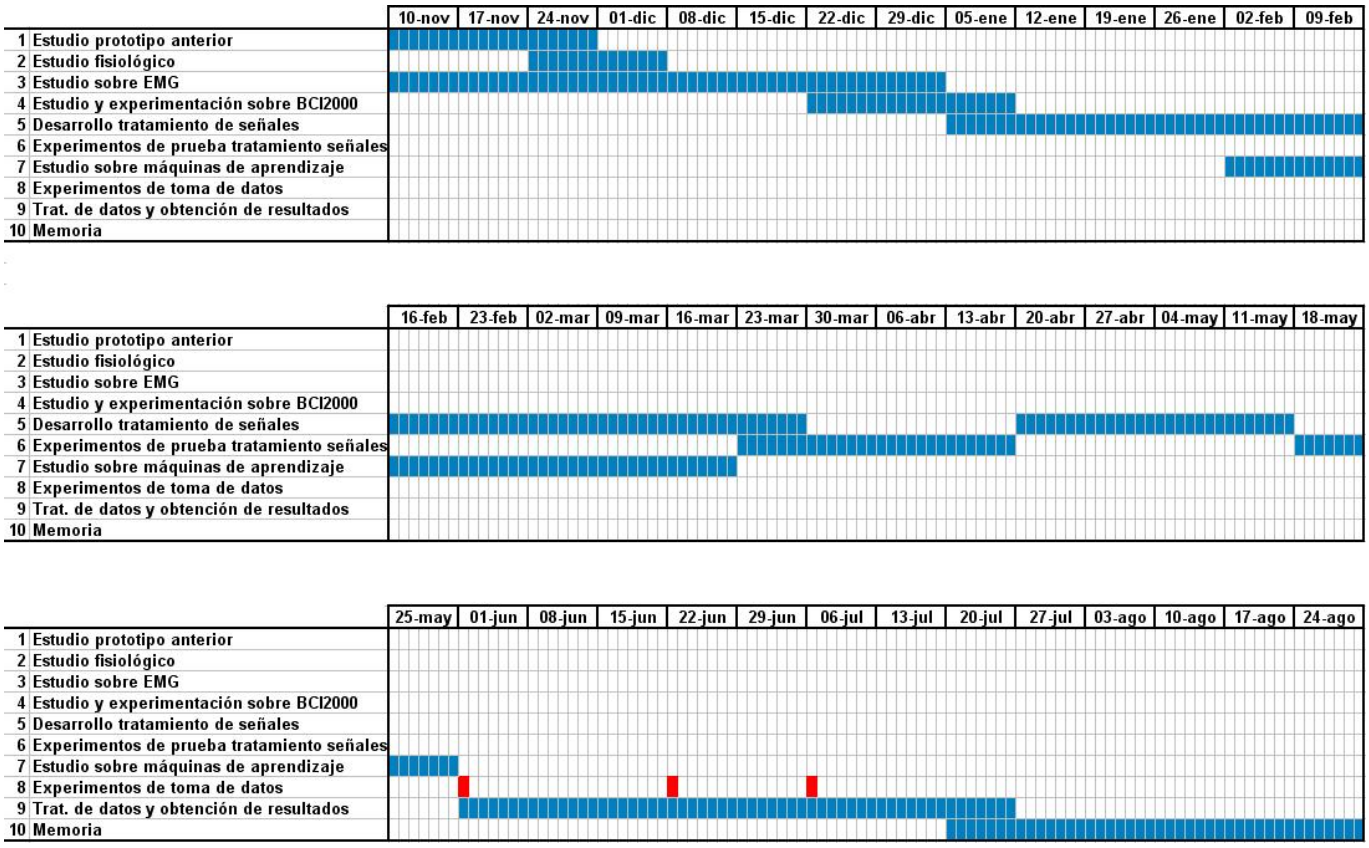


Figura A.1: Diagrama de Gantt del proyecto.

La figura A.1 muestra el diagrama de Gantt donde se observa la distribución temporal, a grandes rasgos, de las tareas realizadas en el presente proyecto. Pese a que en el gráfico aparecen todas las fechas desde principios de noviembre hasta finales de agosto, durante algunas semanas el ritmo de trabajo no ha sido el mismo, ya que, por ejemplo en las franjas

de finales de enero-principios de febrero y final de junio-principio de julio, el autor tuvo que afrontar otras actividades académicas como exámenes o trabajos y alguna semana de julio y agosto no han sido productivas por descanso vacacional.

Como se observa, algunas tareas fueron realizándose en paralelo y es especialmente importante remarcar la cantidad de tiempo empleada en el estudio e investigación de las tecnologías empleadas.

A.3. Empleo de tiempos

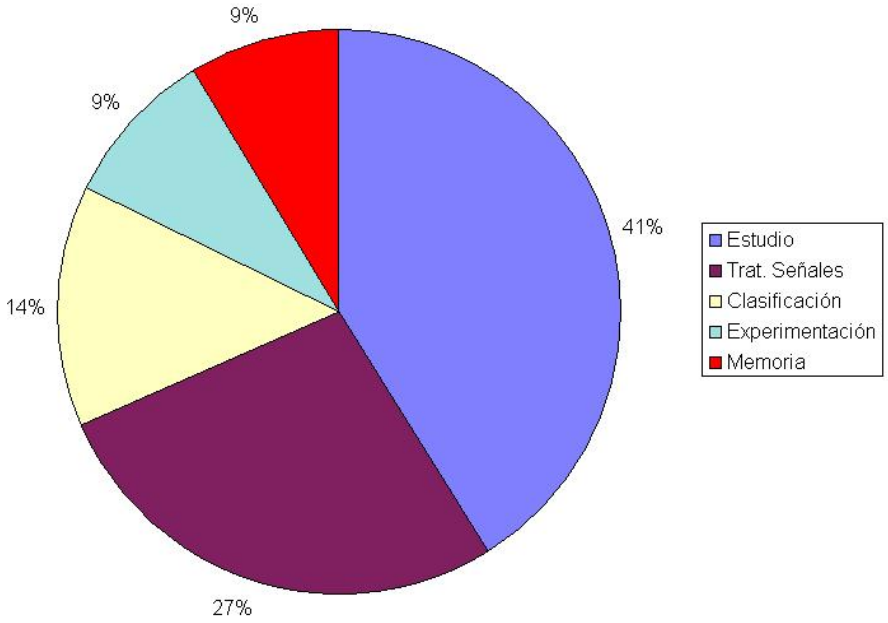


Figura A.2: Distribución global de tiempos.

El cálculo del tiempo requerido para completar este proyecto no era una tarea fácil a priori, ya que era imposible determinar los problemas ajenos al campo informático que podrían surgir. Finalmente, se estima que se han tardado alrededor de 640 horas, dado que de las 42 semanas que han transcurrido desde el inicio al final del proyecto, podrían descontarse unas 10 correspondientes a fechas de exámenes y épocas festivas y se ha trabajado del orden de 4 horas por día de lunes a viernes.

La figura A.2 muestra la división aproximada en porcentaje del tiempo dedicado a cada una de las partes diferentes del proyecto. Como se ve, la mayor parte del tiempo ha ido dedicada a la investigación, seguida por el tratamiento de las señales. La experimentación, pese a ser algo que cuesta menos tiempo proporcionalmente, es una de las partas más importantes y a la que hay que dedicar un mayor cuidado por depender de ella todos los resultados finales.

Por ser éste un proyecto de investigación y, al haberse dedicado la mayor parte del tiempo a esto, se ha desglosado en las distintas partes estudiadas (figura A.3). Como podría esperarse, en lo que más tiempo se ha invertido es en el estudio de la electromiografía, algo que era totalmente desconocido para el autor; y en las máquinas de aprendizaje, un punto clave para la realización de un buen clasificador. Dentro del concepto investigación se incluyen tanto la lectura, como la búsqueda de documentos, así como las pruebas que han tenido que realizarse para aprender a emplear el material utilizado, con todos los clasificadores estudiados, etc.

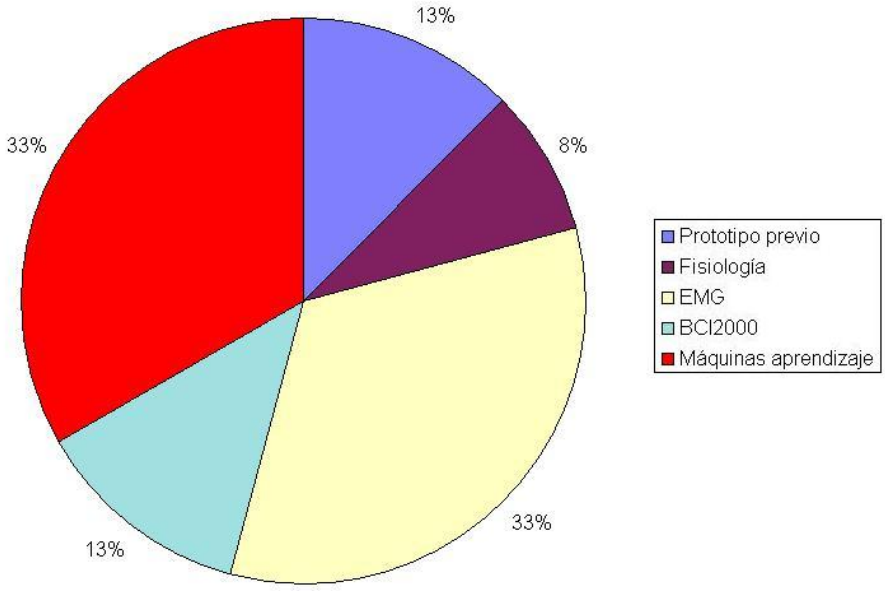


Figura A.3: Distribución del tiempo dedicado a la investigación.

Anexo B. Fisiología del habla

El lenguaje es una herramienta que el hombre ha utilizado como un medio de comunicación desde el comienzo de su existencia. El habla y, en particular, la voz es una vía gracias a la cual los seres humanos han construido una incomparable forma de intercomunicarse que constituye, además, una expresión de la persona en su totalidad y que consiste en el sonido producido por la laringe, haciendo vibrar las cuerdas vocales a partir del aire pulmonar espirado; este sonido es amplificado y reforzado por las cavidades de resonancia.

B.1. Sistema del lenguaje oral

Para entender cómo funciona la comunicación mediante el lenguaje oral hay que conocer los cuatro puntos principales que lo componen: el origen del mensaje, el aparato encargado de transmitirlo, el receptor y, por último, el destinatario.

El origen del mensaje se sitúa en el cerebro, lugar donde la persona que desea comunicarse piensa qué es lo que quiere decir.

El sistema de transmisión está formado por el aparato fonador, el cual lo componen los órganos de respiración (pulmones, bronquios y tráquea), todos ellos infraglóticos[1]; los órganos de fonación o cavidad laríngea (cuerdas vocales); y los órganos de articulación (cavidad nasal y cavidad bucal, formada por labios, dientes, paladar, campanilla y lengua), que son supraglóticos.

El aparato receptor es el oído, lugar donde se recogen las ondas sonoras y encargado de transmitir su información.

Esta información va dirigida al cerebro de la persona receptora, lugar donde se procesa y se decodifica para comprender el mensaje enviado inicialmente.

[1]Situados por debajo de la glotis.

B.2. Alteraciones del lenguaje y del habla

Es importante distinguir entre las alteraciones del lenguaje y las del habla. Las primeras afectan a la creación y comprensión del mensaje, mientras que las segundas atacan a la producción de éste, o lo que es lo mismo, a su realización motora.

Como alteraciones del habla, las principales son la afonía, la mudez y el mutismo, que pese a ser conceptos similares, son cosas diferentes:

- La afonía es simplemente la falta de voz, cosa que puede ser o no temporal. Usualmente se confunde con la *disfonía*, que es una alteración de la voz en la que sí se emiten sonidos.
- La mudez puede referirse a la imposibilidad física de hablar.
- El mutismo es el silencio deliberado y esta variación se manifiesta en enfermos con problemas psicológicos. El tratamiento consiste en ayuda psicológica y el consejo de un patólogo del habla o foniatra.

Por tanto, las aplicaciones de éste prototipo están dirigidas a la afonía permanente y a la mudez, en las que exista conservación de la mímica o gesticulación facial, aunque sea en la mitad de la cara.

Los trastornos que afectan al origen y destino del mensaje, o sea al cerebro, son patologías de base neurológica y no pueden ser solucionados con este sistema. A esos niveles, la alteración del lenguaje que se produce es la llamada *afasia*, ya sea expresiva o receptiva y consiste en un trastorno que deteriora la expresión, la comprensión o la interpretación de la palabra, así como la lectura y la escritura. Las patologías que pueden producir este tipo de trastornos suelen ser los tumores cerebrales, los accidentes cerebrovasculares y los traumatismos craneoencefálicos.

B.3. Causas de la mudez y afonía con conservación de la mímica

Las patologías de base neurológica suelen afectar al cerebro, y las de base fisiológica o anatómica afectan exclusivamente a los órganos del habla y la audición. En estas últimas está fijada la atención de este proyecto. La mudez y la afonía son sólo síntomas que, teniendo una base funcional o anatómica, pueden obedecer a múltiples causas; las principales son físicas y se relacionan principalmente con:

1. **Alteraciones, lesiones o malformaciones del aparato fonador**

- Una causa común es la ruptura del nervio laríngeo recurrente, el cual dirige casi todos los músculos de la laringe. El daño a dicho nervio puede provenir de cirugía (por ejemplo la operación de tiroides) o de un tumor.
- Las personas traqueotomizadas o laringectomizadas.
- Tumores de cuerdas vocales, cavidad nasal y cavidad bucal.
- Tumores o alteraciones de vías respiratorias.
- Otras malformaciones congénitas o adquiridas del aparato fonador, como la parálisis facial unilateral.

2. **Alteraciones del aparato receptor**
   La mudez se asocia, a menudo, con alteraciones del oído que producen sordera. Muchas personas sordas de nacimiento, al no haber oído nunca, no son capaces de aprender a hablar, pese a que consiguen, por mímica, gesticular las palabras.

Anexo C. Electromiografía

C.1. Definición y usos

La electromiografía es una técnica de diagnóstico médico que permite obtener y evaluar las señales bioeléctricas musculares mediante la colocación de unos pequeños electrodos de bajo voltaje en el territorio que se desea estudiar. Las señales adquiridas, o electromiograma, oscilan entre los $50\mu V$ los $30mV$, dependiendo del tamaño del músculo (cuanto más grandes sean, mayores son los potenciales provocados). El tejido muscular es eléctricamente neutro cuando se encuentra en reposo, sin embargo al expandirse o contraerse se producen unos patrones de conducción nerviosa que tienen su origen en las membranas de las células que forman esos músculos.

Existen dos formas de adquisición de EMG: métodos invasivos, consistentes en la utilización de electrodos de aguja, que se insertan en las fibras musculares atravesando la piel y métodos no invasivos o EMG superficial, para el que únicamente se requiere que el sensor permanezca en contacto con la superficie de la piel. La primera técnica es utilizada para pruebas médicas que requieren la observación de fibras musculares concretas, esto puede detectar actividad espontánea en alguna zona, lo que podría estar causado por algún daño en músculos o nervios. Por otro lado, existen pruebas para las que una técnica invasiva puede no ser necesaria, ya que no se requiere la exactitud de medir solamente unas pocas fibras; en ese caso se realizan adquisiciones superficiales, que medirían con suficiente precisión activaciones musculares o movimientos en los que se impliquen varios músculos.

C.2. Infraestructura utilizada

Esta sección mostrará la infraestructura que se ha utilizado para la adquisición de las señales EMG.

C.2.1. Electrodos

Los sensores utilizados para la adquisición de datos son unos electrodos de la marca Grass Technologies. Con forma de disco y un agujero por el que puede ser introducido el de gel conductor con una jeringuilla, están fabricados en oro, con un diámetro de 10mm. Pueden verse en la imagen C.1.

Figura C.1: Electrodos Grass.

C.2.2. Amplificador

El amplificador empleado para conectar los electrodos con el computador en las sesiones de experimentación realizadas es el que aparece en la figura C.2. Posee 16 entradas que permiten adquirir señales con una frecuencia de sampleo de hasta 38.400 Hz por canal. Si se desea, pueden apilarse varios amplificadores iguales para conseguir un sistema de adquisición de 32, 48, 64 o 80 canales. Su comunicación con el computador se realiza mediante una interfaz USB 2.0.

Figura C.2: Amplificador utilizado.

C.2.3. Gel

El gel conductor es de vital importancia para que, al colocarse entre el sensor y la piel del paciente, mejore la conductividad y permita adquirir unas señales mejores.

C.2.4. Otros

Además del material mencionado, es necesario contar en el inventario con algodón y alcohol para limpiar la superficie de la piel, *bastoncillos* con punta de algodón para impregnar los electrodos con el gel y algún tipo de esparadrapo o *tape* con un adhesivo lo bastante fuerte como para que actúe bien de sujeción para los sensores y no se despegue con el sudor del paciente.

C.3. Protocolos de montaje y limpieza

El protocolo de montaje incluye todas las acciones que deben realizarse para preparar al sujeto de pruebas y al sistema para una correcta adquisición de la actividad electromiográfica. El proceso completo puede llegar a costar entre 30 minutos y una hora, pero es muy importante realizarlo de manera cuidadosa para evitar fallos que pueden llegar a causar que las señales adquiridas sean inutilizables.

Antes de comenzar con la colocación de los sensores debe comprobarse que el monitor por el que se mostrarán los estímulos está situado a una distancia correcta, que no fuerza a que la persona ahí sentada tenga una postura en la que cualquier músculo implicado pueda ver afectado su movimiento para las pronunciaciones.

Una vez hecho esto se procede a colocar los sensores por parejas. Para ello se limpia bien la superficie de la piel con algodón impregnado en alcohol, se unta gel conductor en cada electrodo y se sitúan en el músculo deseado, de manera paralela a las fibras de éste (figura 3.3). Después se pone una tira de esparadrapo sobre ellos para que queden bien sujetos.

Los electrodos de tierra y referencia es recomendable ponerlos en primer lugar para así poder ir comprobando las señales adquiridas por cada nuevo canal colocado. Con respecto al resto de canales, la experiencia nos ha enseñado que lo mejor es empezar a montar por los que se vayan a situar más abajo y continuar hacia arriba, para que así no molesten los cables que quedan colgando.

Un aspecto a tener en cuenta y que es muy común pasar por alto es que las sesiones de experimentación deben realizarse en un ambiente idóneo. La sala debe estar lo más despejada posible para agilizar los movimientos del personal, libre de ruidos para no influir en el sujeto y por último y más importante, la temperatura ambiente debe ser

adecuada. El sudor provocado por temperaturas superiores a unos 25ºC puede causar que el esparadrapo no sea capaz de sujetar los electrodos y que sea imposible hacer una buena obtención de datos. Por contra, una temperatura demasiado baja podría hacer que el movimiento de los músculos no sea tan ágil como lo sería en condiciones normales, lo que perjudicaría a las señales adquiridas.

Al finalizar el experimento debe limpiarse cada electrodo individualmente con un cepillo de dientes o similar para eliminar por completo los restos de gel que puedan quedar.

Anexo D. Extracción de características

Las características calculadas sobre las señales EMG, como ya se indicó en la sección 4.3, son 11. Aquí detallaremos qué es exactamente cada una de ellas.

D.1. FFT

La transformada rápida de Fourier (FFT) es un algoritmo que permite calcular, de manera eficiente la transformada discreta de Fourier (DFT). El coste computacional de esta segunda transformada es del orden de n^2operaciones, siendo n el numero de muestras que componen la señal. Aplicando la transformación rápida, se puede conseguir prácticamente el mismo resultado con sólo O(n log n) operaciones.

Por su parte, la DFT es una transformada específica de Fourier empleada en el tratamiento de señales. Se utiliza para analizar las frecuencias presentes en una señal muestreada pasando, para ello, una función del dominio del tiempo al dominio de frecuencias. Requiere una función discreta y finita como entrada, como las señales EMG empleadas en este proyecto, y solamente evalúa los componentes de frecuencia necesarios para reconstruir el segmento finito que se desea evaluar. La transformada inversa no puede reconstruir una señal infinita en el tiempo a no ser que sea periódica, por eso se suele decir que es una transformada para el análisis de Fourier de funciones discretas en el tiempo y con dominio finito.

Por defecto se toman los 20 primeros valores de esta característica, pero se puede cambiar para escoger el número de valores que se desee.

D.2. Downsampling

El subsampleo o downsampling es el proceso de reducir la frecuencia de muestreo de una señal para decrementar el número de datos que la componen. Su uso ha permitido realizar una comparativa clasificando los ejemplos utilizando las señales en crudo submuestreadas y utilizando vectores de características, lo que requiere un procesado más

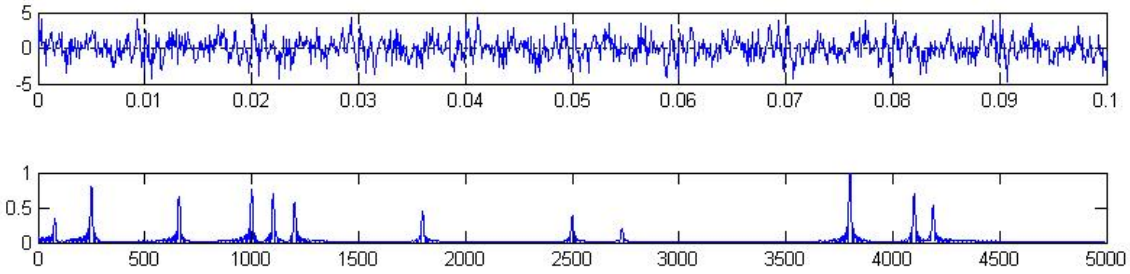


Figura D.1: La gráfica de arriba corresponde a una señal finita en el tiempo entre 0 y 0.1. La de abajo es su transformada de Fourier y, como se puede ver, está en el dominio de frecuencias entre 0 y 5.000.

complejo. Probar a clasificar utilizando toda la señal sin subsamplearla sería inviable, ya que, como se indicó en la sección 4.2.2, las muestras son adquiridas a 2.400 Hz, lo que supone 2.400 valores por cada una. El valor al que se desea realizar el sampleo puede variarse para buscar el valor óptimo. En este trabajo se estudiaron los valores de 40, por ser del mismo orden que el número de valores incluidos en los vectores de características (41); y 80, por ser el doble de éste. Sin embargo, como muestran los resultados en el anexo F, estos submuestreos no son una buena característica con la que clasificar.

D.3. RMS

El Root Mean Square (RMS), también conocido como la media cuadrática, es una medida estadística de la magnitud de una variable, especialmente útil cuando la variable cambia entre valores positivos y negativos.

El cálculo del RMS de una serie de n valores $\{x_1, x_2...x_n\}$ es:

$$x_{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$$

D.4. Amplitud

La amplitud es la magnitud de cambio en una variable oscilante. Las unidades en las que se mide son los voltios, sin embargo, las señales EMG son del orden de los micro voltios. Se han elegido los valores máximo y medio de las señales por ser los que pueden resultar más representativos; el mínimo se ha desechado porque, al tomarse en valor absoluto, en la mayoría de los casos es 0. Las dos amplitudes tomadas (máxima y media) nos darán el máximo valor que toma una muestra en la ventana de tiempo en la que se trabaja y la media de ésta, respectivamente.

También se han tomado como características la suma de todos los valores de la señal en crudo y de la señal rectificada, ya que, pese a que pueden resultar redundantes, son medidas que llevan información valiosa sobre la energía de la señal y, en los tests realizados dieron buen resultado, siendo seleccionadas entre las características más significativas para diferenciar entre ciertas sílabas.

D.5. Kurtosis

La Kurtosis es una medida estadística del apuntamiento, o lo *picuda* que es una distribución, es decir, estudia la concentración de frecuencias alrededor de la media y en la zona central de la distribución. Su cálculo se realiza con la fórmula:

$$Kurtosis = \frac{E(x-\mu)^4}{\sigma^4},$$

donde x es la señal, μ es la media de ésta, σ es su desviación estándar y E es el valor esperado. Una distribución normal tiene una kurtosis igual a 3 (distribución mesocúrtica), si su valor es mayor se trata de una distribución más apuntada que la normal (leptocúrtica), si es menor que 3, será menos apuntada (platicúrtica).

D.6. MFCC

Los coeficientes cepstrales en las frecuencias de Mel (Mel-frequency cepstral coefficients o MFCC) son unos coeficientes para la representación del habla, basados en la percepción auditiva humana. Similar a la transformada de Fourier, la diferencia básica entre éstas, es que en MFCC las bandas de frecuencia se sitúan de manera logarítmica, lo que modela más apropiadamente la respuesta auditiva humana. Pese a que por su naturaleza se trata de características empleadas en el reconocimiento de voz, también son utilizadas con buenos resultados en reconocedores del habla mediante EMG ([9], [27]).

D.7. IAV

El Integrated Absolute Value o Valor Absoluto Integrado es una estimación del valor absoluto medio de una señal tomada por segmentos. El valor absoluto de cada segmento i, que contenga N muestras, se define como:

$$\bar{X}_i = \frac{1}{N}\sum_{k=1}^{N}|x_k|$$

D.8. Zero Crossing

Esta característica calcula el número de veces que la señal cambia su valor de positivo a negativo, o viceversa. Al ser las señales EMG tan ruidosas y, para evitar que el cálculo quede enturbiado por las grandes variaciones que produce ese ruido, se filtran antes del cálculo, haciendo que los valores entre -25 y +25 μV tomen el valor 0; así sólo se contabilizarán como cruces por cero los cambios de signo desde amplitudes superiores o inferiores a estos valores.

Anexo E. Clasificación

En este anexo se definirá qué es un problema de clasificación, sus tipos y se detallarán en mayor medida algunos de los conceptos mencionados en el texto relacionados con este tema.

E.1. El problema de clasificación

Para explicar qué es un problema de clasificación la mayoría de autores recurren a ejemplos que pongan al lector en situación ([21],[22],[23]). El objetivo principal de un problema de este tipo es, dada una serie de observaciones de muestras pertenecientes a distintas clases, ser capaz de clasificar una nueva muestra no observada previamente. En la literatura disponible en inglés se le denomina de varias maneras: *statistical learning*, *discrimination*, *machine learning*, *pattern recognition*, etc.

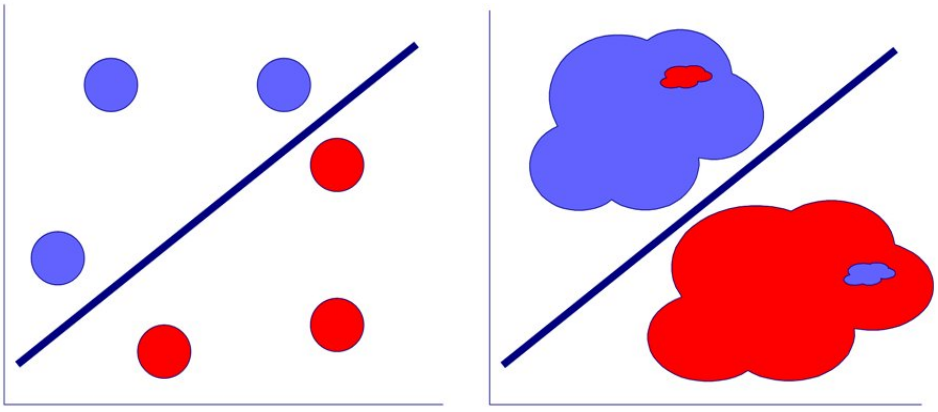
Según el número de clases, podemos referirnos a un problema como multiclase, donde para cada muestra habrá que discriminar a qué grupo corresponde, o uniclase en el que habrá que determinar si pertenece o no a una clase dada (sección E.2). También, según la naturaleza del problema podemos encontrarnos con un problema de clasificación con aprendizaje supervisado o no supervisado (sección E.3) dependiendo de si se conocen o no las clases a las que pueden pertenecer las muestras.

Las herramientas para su resolución se denominan clasificadores y son tan variadas como variados son los problemas que existen. Algunos de los más utilizados son las redes neuronales, las máquinas de soporte vectorial, los árboles de decisión o los clasificadores Bayesianos (sección E.6).

E.2. Problemas uniclase vs. multiclase

El problema uniclase más sencillo es aquél que puede ser resuelto con un clasificador lineal. Como puede verse en la figura E.1(a), hay dos clases, una roja y una azul, que pueden ser separadas fácilmente por una línea recta. La denominación *uniclase* dada

al problema es porque podríamos plantearlo de manera que, para cada nueva muestra, deberíamos determinar si pertenece o no (*true/false*) a la clase azul o si pertenece o no a la clase roja. Construir un clasificador de este tipo no es difícil, ya que solamente habría que definir la ecuación de la recta que es capaz de dividir las dos clases para obtener un 100 % de acierto en el reconocimiento. Pero en el mundo real, los problemas con los que hay que enfrentarse no suelen ser tan sencillos, y se aproximan más a lo que modela la figura E.1(b), donde con un clasificador lineal no es posible distinguir totalmente entre las dos clases y no se alcanzarán porcentajes de acierto del 100 %, aunque si son pocas las muestras que no se reconocen correctamente, la solución puede darse por buena.



(a) (b)

Figura E.1: Dos problemas con dos clases separadas por un clasificador lineal.

Sin embargo, ni siquiera todos los problemas con dos clases pueden ser separados por un clasificador lineal. La figura E.2 muestra el clásico problema de la función XOR (OR exclusivo) donde con una sola recta es imposible distinguir entre las dos clases y se requerirían varios clasificadores para conseguir solucionarlo.

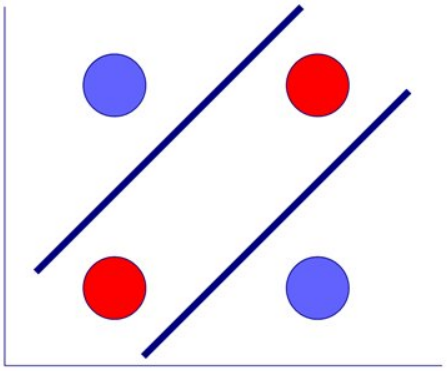


Figura E.2: Problema de clasificación XOR.

El siguiente paso son los problemas multiclase, en el que no solo tenemos que distinguir si una muestra pertenece o no pertenece a una clase, sino que existen varias opciones,

de las cuales hay que conseguir adivinar a cuál corresponde (figura E.3). Construir un clasificador de este tipo es más complicado, pero podría realizarse utilizando varios clasificadores uniclase en paralelo, que proporcionasen como resultado, además de si la muestra corresponde o no a una clase, un porcentaje aproximado, calculado por el clasificador, de fiabilidad de acierto. Con lo cual, una vez todos los clasificadores dan su resultado con probabilidad, se escogería la clase que tiene un valor más alto.

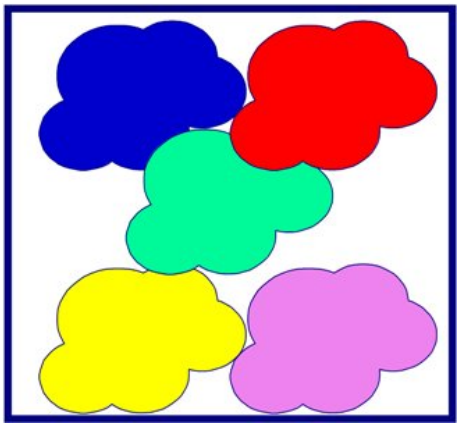


Figura E.3: Problema de clasificación multiclase.

E.3. Aprendizaje supervisado vs. no supervisado

El problema principal en el que se ha centrado este proyecto ha sido el de diferenciar muestras de señales EMG correspondientes a 30 sílabas distintas. Para solucionarlo se diseña una máquina de aprendizaje y se entrena con ejemplos de los que se conoce con certeza a qué grupo pertenecen. Una vez se ha entrenado, se pueden introducir nuevos ejemplos y el clasificador determinará a qué clase se parece mas. Esto es lo que se conoce como aprendizaje supervisado, que el clasificador al principio tenga una información sobre las clases que tiene que tratar de discriminar.

Sin embargo, existen problemas donde no existe esa información. Desde el principio se entregan a la máquina una serie de ejemplos sin más que sus atributos y ésta tendrá que agruparlos buscando similaridades entre ellos. El aprendizaje no supervisado funciona mediante algoritmos de agrupamiento o *clustering*. Una de las mayores dificultades que presentan los sistemas de aprendizaje no supervisado es definir el concepto de *similaridad* entre dos vectores de características, ya que dependiendo de esto se producirán los agrupamientos de una u otra manera.

E.4. True positives, true negatives, false positives, false negatives

Conocer el significado de estos cuatro conceptos es clave para entender los resultados obtenidos en cualquier problema de clasificación. Para comprenderlos fácilmente tomaremos como ejemplo un problema uniclase sencillo en el que si un ejemplo dado pertenece a la clase objetivo diremos que es positivo (P), si no pertenece, será negativo (N).

Una vez planteado el problema habría que construir una máquina de aprendizaje que tratase de modelarlo, entrenarla y clasificar con ella el conjunto de ejemplos disponible. Con esto hecho se obtendría como resultado una matriz de confusión como la que aparece en la tabla E.1.

	P	N
P	*TP*	*FN*
N	*FP*	*TN*

Tabla E.1

Los *true positives* (*TP*) corresponden al número de ejemplos que, siendo positivos, han sido clasificados por la máquina como positivos. Los *true negatives* (*TN*) son los que, siendo negativos, han sido clasificados como negativos. Estos dos valores son los que se desea maximizar en mayor medida, ya que determinan lo bueno que es el reconocedor.

Los *false positives* (*FP*) son ejemplos que son negativos pero que la máquina ha determinado que son positivos, mientras que los *false negatives* (*FN*) son muestras negativas que han sido clasificadas como positivas.

Utilizando estos valores se puede medir el rendimiento o *accuracy* del sistema con la siguiente fórmula:

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN}$$

Sin embargo esta medida puede ser poco indicativa del verdadero poder de reconocimiento si se dispone de un clasificador que distingue muy bien los *true negatives* pero no los *true positives*. Así, si se introducen muchos ejemplos negativos el rendimiento calculado sería muy alto pero, normalmente, lo que se quiere obtener es una buena identificación de los positivos. Por eso, la medida del rendimiento suele acompañarse con las de sensibilidad y especificidad:

$$Sensitivity = \frac{TP}{TP + FN},$$
$$Specificity = \frac{TN}{TN + FP},$$

que proporcionan los porcentajes de *TP* sobre el total de los ejemplos positivos y *TN* sobre el total de los negativos, respectivamente.

E.5. Cross-validation

La utilización de la técnica *cross-validation* surge del hecho de que no siempre se dispone de un número lo bastante grande de ejemplos con los que entrenar una máquina de aprendizaje. Normalmente, del conjunto total de muestras disponibles se escoge al azar un porcentaje (70, 80, 90 % suelen ser los más utilizados) y se utiliza para entrenar, el resto se clasifica para obtener unas medidas de rendimiento. Pero, como es natural, podría ocurrir que los datos escogidos para entrenar (o clasificar) no fuesen representativos y, por tanto, el resultado final no fuese realista.

Una forma para mitigar este problema sería repetir el proceso varias veces tomando al azar nuevos datos para entrenar y clasificar. Para ser más rigurosos, se puede incluso hacer que todos los ejemplos se utilicen el mismo número de veces para cada tarea. Ésa es, básicamente la idea de esta técnica: *X fold cross-validation* consiste en dividir el total de datos en *X* grupos de igual tamaño, utilizar todos menos uno para entrenar y el restante para clasificar. Después se calcula la media de los porcentajes obtenidos y así se consigue un resultado mucho más exacto del que se obtendría con cualquier división aleatoria.

Varios tests han demostrado que la utilización de 10 divisiones proporciona unos resultados más exactos en las estimaciones de acierto-error [23]. Por eso en todas las pruebas realizadas en este trabajo se ha empleado esta técnica.

E.6. Métodos de clasificación

La clasificación es una rama de la inteligencia artificial que se basa en el reconocimiento de patrones. Se trata de imitar el funcionamiento del cerebro humano, haciendo que primero se aprenda a partir de una serie de ejemplos, para después conseguir un razonamiento que produzca un resultado. En esta sección va a describirse qué son y cómo funcionan algunos de los métodos de clasificación más utilizados.

E.6.1. Árboles de decisión

Un árbol de decisión es una herramienta de clasificación que se representa con forma de árbol y que modela decisiones y las posibles consecuencias de cada una de éstas (figura E.4). Es una solución ideal para problemas multiclase, los cuales va dividiendo para facilitar su resolución.

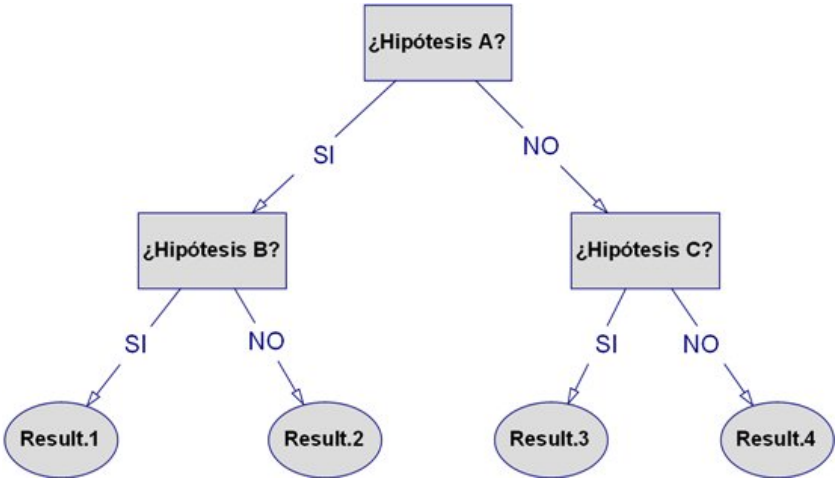


Figura E.4: Árbol de decisión.

Para cada nuevo ejemplo presentado, el árbol irá aplicando los tests correspondientes a cada nodo y, según el valor de los atributos correspondientes, seguirá unas u otras ramas hasta alcanzar un nodo hoja, que determinará el resultado final. Lo más complicado es la construcción del árbol y la generación de reglas a partir de éste, ya que la clasificación de nuevas muestras es prácticamente inmediata. Lo que realmente define cada modelo de árbol de decisión es la forma en la que se construye, algunos de los más utilizados son ID3, C4.5 y su última versión C5.0.

E.6.2. Clasificación Bayesiana

El primer concepto que hay que explicar en este apartado es el teorema de Bayes, ya que es la piedra angular de todos los clasificadores Bayesianos[22], para ello representaremos como *P(a)* la probabilidad de que ocurra a, y como $P(a|b)$ la probabilidad de que ocurra a dado b. Con esto dicho, el teorema es el siguiente:

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)},$$

de lo que fácilmente deducimos que $P(a|b)$ incrementa su valor conforme se incrementan $P(a)$ y $P(b|a)$, y se decrementa al crecer $P(b)$.

Para conectar esto con los problemas de *machine learning* tomaremos b como el conjunto de datos de entrenamiento y a como una de las clases pertenecientes al universo de clases total. Por consiguiente, buscaremos cuál es la probabilidad de que una muestra tomada pertenezca a la clase a dado el conjunto de entrenamiento b.

El clasificador Naive Bayes aplica esto a las tareas de entrenamiento, donde a cada instancia x se la describe a partir de un conjunto de atributos y donde la función objetivo

f(x) puede tomar cualquier valor (v_j) de los pertenecientes al universo de clases (V). Dados una serie de datos de entrenamiento y una nueva muestra definida por la tupla de atributos (a_1,a_2...a_n), el clasificador tratará de predecir la clase a la que esa muestra pertenece. Para ello le asignará la que posea probabilidad mayor, v_P, dados los atributos que la componen.

$$v_P = \text{máx}\, P(v_j|a_1, a_2...a_n)$$

Utilizando el teorema de Bayes, podemos reescribir la ecuación anterior de la siguiente manera:

$$v_P = \text{máx}\, \frac{P(a_1, a_2...a_n|v_j)P(v_j)}{P(a_1, a_2...a_n)} = \text{máx}\, P(a_1, a_2...a_n|v_j)P(v_j)$$

Por consiguiente, hay que estimar los dos términos basándose en los datos de entrenamiento. Estimar cada uno de los $P(v_j)$ es fácil, simplemente contando el número de ocasiones en las que aparece v_j en el subconjunto de entrenamiento. Sin embargo, estimar los distintos términos $P(a_1, a_2...a_n|v_j)$ es complicado si no se dispone de un conjunto muy elevado de datos para entrenar. El problema de esto es que el número de términos es igual al número de posibles instancias multiplicado por el número de clases, por consiguiente hay que revisar cada muestra tantas veces como grupos distintos existan.

El clasificador Naive Bayes se basa en asumir como independientes los valores de los atributos dada una clase, lo que es lo mismo, se asume que dada una posible clase para una instancia, la probabilidad de tener la conjunción $a_1, a_2...a_n$ es el producto de todas las probabilidades individuales para cada atributo: $P(a_1, a_2...a_n|v_j) = \prod_i P(a_i|v_j)$. Sustituyendo esto en la ecuación obtenida previamente se consigue la fórmula del clasificador Naive Bayes:

$$v_{NB} = \text{máx}\, P(v_j) \prod_i P(a_i|v_j)$$

Donde v_{NB} representa la clase que Naive Bayes da como salida. Notar que el número de términos $P(a_i|v_j)$ a calcular a partir del conjunto de entrenamiento es igual al número de valores de atributos distintos por el número de clases, un número mucho menor que si hubiese que calcular todos los $P(a_1, a_2...a_n|v_j)$ mencionados anteriormente.

E.6.3. Boosting

El Boosting es un meta-algoritmo que trata de mejorar el rendimiento obtenido por un sistema de aprendizaje supervisado. Kears y Valiant ([28]) propusieron en 1988 que un conjunto de clasificadores débiles (weak classifiers) podían ser combinados para formar

un clasificador fuerte (strong classifier), que supere su rendimiento, mientras que en 1990, Schapire ([29]) demostró matemáticamente que a cualquier algoritmo de aprendizaje débil se le puede aplicar un Boosting para formar un algoritmo de aprendizaje fuerte.

El método consiste en ejecutar iterativamente los clasificadores débiles y asignarles distintos pesos a cada uno. A los datos de entrenamiento se les asignan también unos pesos, inicialmente idénticos, y tras cada iteración se reajustan, asignando valores mayores a las muestras que han quedado clasificadas incorrectamente y valores menores a las que se clasificaron bien. Así, en cada iteración, el clasificador débil buscará la manera de agrupar correctamente los ejemplos que tienen un mayor peso. Finalmente, según las muestras correctas que es capaz de reconocer cada clasificador débil, tendrá un mayor o menor peso en la formación del clasificador fuerte, lo que hará que la decisión tomada por cada algoritmo débil se tenga más o menos en cuenta.

E.6.4. Redes neuronales

Las redes neuronales artificiales (ANN, del inglés Artificial Neural Networks) son un paradigma de aprendizaje que trata de imitar el comportamiento de las neuronas que forman el sistema nervioso animal. Funciona interconectando neuronas entre sí para que formen una red que produzca una salida conjunta. Útiles para resolver problemas no lineales, suelen formar un sistema adaptativo, capaz de cambiar su estructura basándose en información que fluye a través de la red durante el proceso de aprendizaje.

Como muestra la figura E.5, suelen estructurarse por capas independientes entre sí, siendo lo más común que tengan una capa de entrada que reciba los datos, una oculta encargada del grueso del procesamiento y una de salida, que genere el resultado.

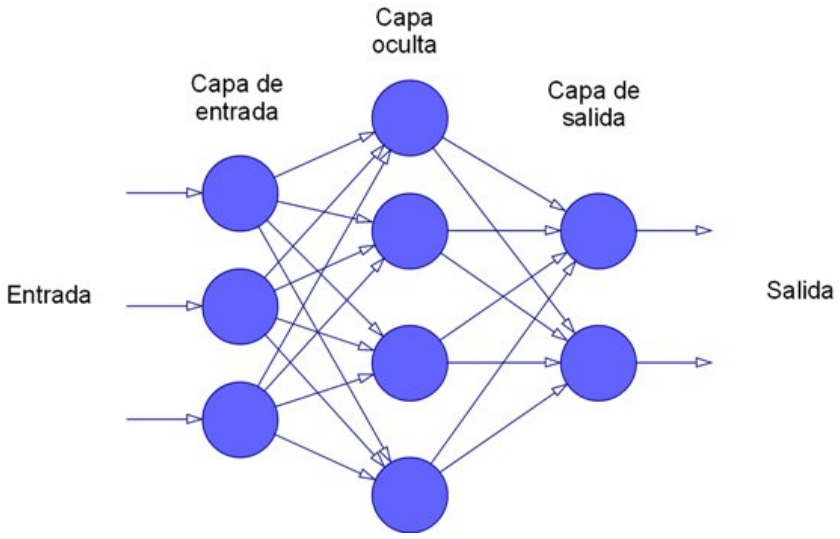


Figura E.5: Red neuronal.

E.6.5. Máquinas de soporte vectorial

Las máquinas de soporte vectorial comprenden una familia de métodos de aprendizaje supervisado utilizados para clasificación y regresión. Tomando los datos de entrada como dos conjuntos de vectores en un espacio n-dimensional, proponen una solución lineal empleando hiperplanos para separar ese espacio. Calculan aquél que consigue maximizar el margen entre los dos vectores, para ello se construyen varios y se busca cuál es capaz de distinguir mejor entre las dos clases. Como es natural, será aquél que guarde la distancia máxima entre los ejemplos que las componen (figura E.6).

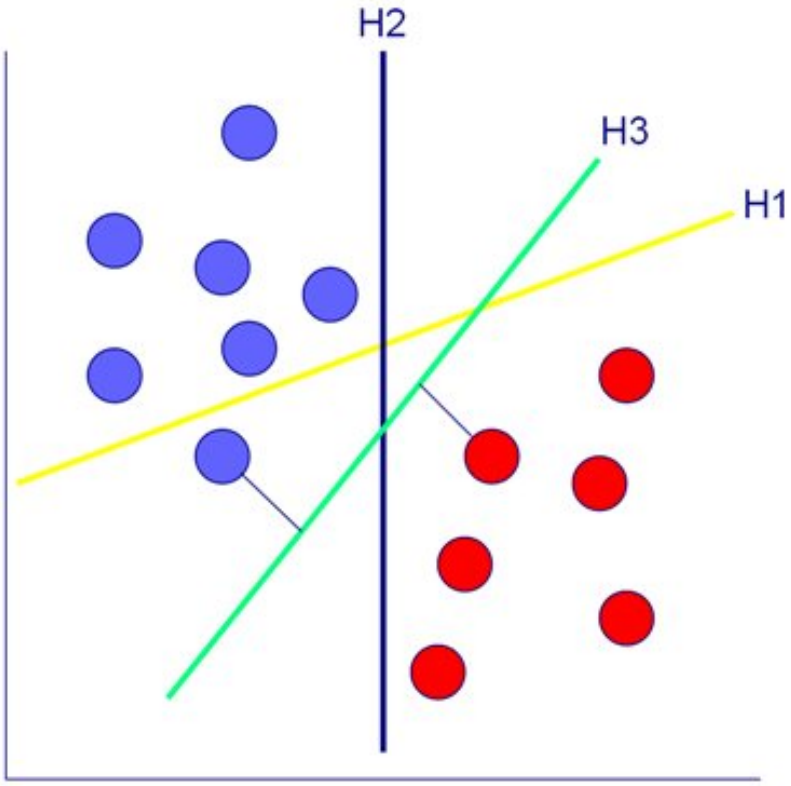


Figura E.6: El Hiperplano H1 no separa bien entre las dos clases; H2 las separa pero con un margen pequeño; H3 lo hace con el mayor margen posible.

E.7. El formato .ARFF

El formato .ARFF es un estándar utilizado por el software Weka para estructurar los datos que utiliza como entrada. La figura E.7 muestra un ejemplo de un archivo con este formato correspondiente a un problema de toma de decisión sobre si jugar o no un partido de tenis, dependiendo de las condiciones del tiempo. Posee cinco atributos, entre numéricos y nominales, y 14 muestras.

La estructura es muy sencilla: las palabras clave *relation*, *attribute* y *data* deben ir precedidas por el símbolo @, los atributos se separan entre sí por comas y los comentarios deben ir precedidos por un %. Para la definición de atributos nominales deben enumerarse todos los posibles valores entre llaves y para los numéricos basta con la palabra clave *numeric*.

```
% ARFF file for the weather data with some numeric features
%
@relation weather

@attribute outlook { sunny, overcast, rainy }
@attribute temperature numeric
@attribute humidity numeric
@attribute windy { true, false }
@attribute play? { yes, no }

@data
%
% 14 instances
%
sunny, 85, 85, false, no
sunny, 80, 90, true, no
overcast, 83, 86, false, yes
rainy, 70, 96, false, yes
rainy, 68, 80, false, yes
rainy, 65, 70, true, no
overcast, 64, 65, true, yes
sunny, 72, 95, false, no
sunny, 69, 70, false, yes
rainy, 75, 80, false, yes
sunny, 75, 70, true, yes
overcast, 72, 90, true, yes
overcast, 81, 75, false, yes
rainy, 71, 91, true, no
```

Figura E.7: Ejemplo de fichero ARFF tomado del libro [23].

Anexo F. Resultados

F.1. Clasificaciones de vocales

Para construir un clasificador que reconozca un número medio-alto de clases (en nuestro caso 30), debemos empezar dividiendo el problema y, para ello, emplearemos solamente las 5 vocales para buscar el clasificador más adecuado y determinar qué características proporcionan mejores resultados. Todos los resultados mostrados en esta sección corresponden a pruebas realizadas con 200 muestras de cada vocal, obtenidas en una misma sesión de adquisición y clasificadas validando con *10 fold cross-validation*.

F.1.1. Comparación de características

En este apartado vamos a utilizar 4 métodos de clasificación distintos: Naive Bayes, árbol de decisión J4.8, AdaBoost combinado con Naive Bayes y Adaboost combinado con el árbol J4.8. Estos nos servirán para comparar las clasificaciones tomando como datos los submuestreos de la señal y los vectores de características.

F.1.1.1. Downsampling 40 Hz

En la figura F.1 vemos una comparativa de los resultados obtenidos con los diferentes clasificadores. En el eje X mostramos los clasificadores con y sin combinar con AdaBoost, mientras que en el eje Y observamos el porcentaje de acierto en la clasificación.

Como podemos observar, los porcentajes obtenidos no son demasiado buenos, sin embargo, todos ellos son superiores al resultado que obtendríamos con un clasificador que diese un resultado aleatorio para cada muestra, ya que, en ese caso, al ser éste un problema de 5 clases, acertaría en el 20 % de los casos. El porcentaje de *true positives* obtenido con el árbol de decisión J4.8 es del 48,2 %, el que nos proporciona Naive Bayes es 56,4 %. Ambos mejoran al ser combinados con AdaBoost, ejecutándose 10 iteraciones, hasta el 56 % y 60 % respectivamente. Por tanto, para este caso determinamos que Naive Bayes, ejecutándose con AdaBoost es el mejor clasificador.

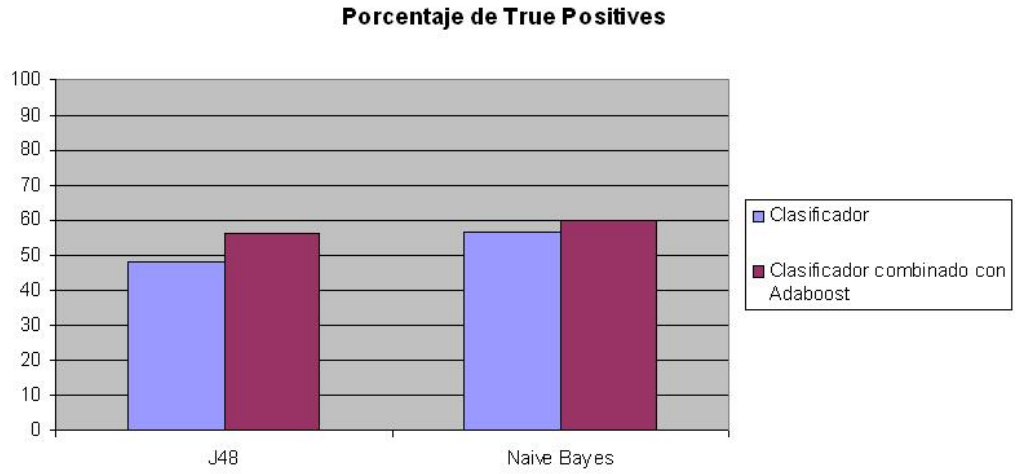


Figura F.1: Comparativa de resultados de clasificación obtenidos para 4 máquinas de aprendizaje distintas, usando como característica el downsampling a 40Hz.

F.1.1.2. Downsampling 80 Hz

El gráfico de la figura F.2 muestra, igual que en el apartado anterior, la comparativa entre los distintos clasificadores. Como puede verse, la mejora entre utilizar 40 y 80 muestras por señal es mínima, ya que en ninguno de los casos llega a mejorar ni siquiera un 5 %, en contraposición, el tamaño y el tiempo necesario para almacenar y entrenar las muestras es el doble.

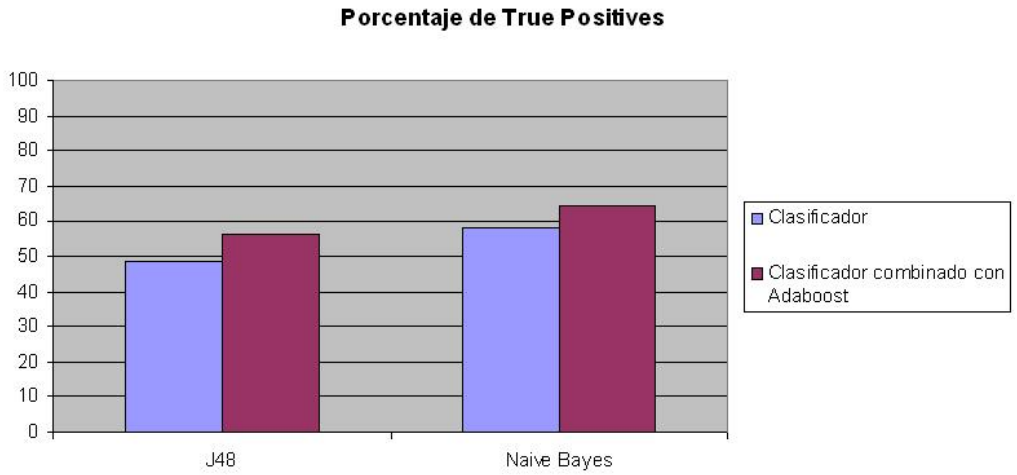


Figura F.2: Comparativa de resultados de clasificación obtenidos para 4 máquinas de aprendizaje distintas, usando como característica el downsampling a 80Hz.

F.1.1.3. Clasificaciones con vector de características

En este punto se muestran los resultados obtenidos al usar para la clasificación los vectores compuestos por las características implementadas, explicadas en el anexo D (a excepción del downsampling).

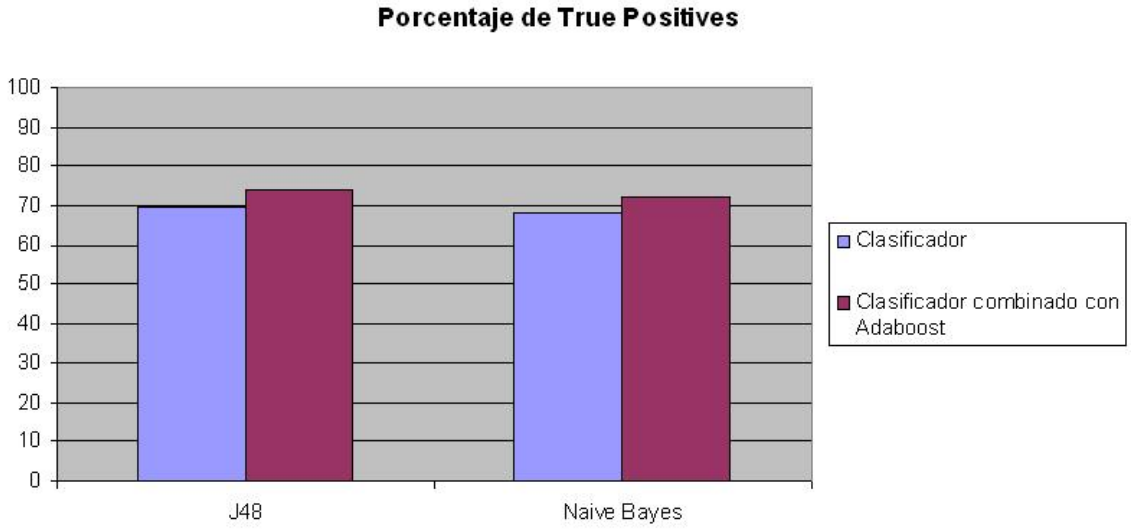


Figura F.3: Comparativa de resultados de clasificación obtenidos para 4 máquinas de aprendizaje distintas, usando vectores de características.

Puede observarse claramente que el porcentaje de acierto es mucho mayor utilizando un procesamiento inteligente que clasificando con las señales en crudo. Los resultados obtenidos superan en un número próximo al 20 % a los que proporcionaba el downsampling. Para este caso, además, se comprueba que es mejor clasificador AdaBoost combinado con el árbol de decisión J4.8 que con Naive Bayes, aunque sea por poco ($\approx$ 2 % más de *true positives*). Esto puede deberse a que Naive Bayes, por definición, toma todos los valores de las características como independientes entre sí y eso puede ser válido en vectores de downsampling por el ruido que contienen, pero para las características correspondientes a transformadas y demás operaciones, los valores están más correlacionados entre sí y, tratarlos como independientes, es algo que elimina información útil. Por eso el árbol de decisión acierta en mayor medida y, por eso, será el clasificador que se utilizará en el resto de comparativas.

Una vez decidido el clasificador que se va a emplear en los futuros esquemas, se decide probar iterar 100 veces con el árbol de decisión para obtener un mejor resultado, con lo que finalmente se consigue un 80,2 % de aciertos. En la tabla F.1 aparecen las matrices de confusión obtenidas para las vocales utilizando AdaBoost con el árbol J4.8 iterando 10, 50 y 100 veces. Las mejoras obtenidas al pasar de 50 a 100 iteraciones no son muy grandes comparadas con el tiempo extra necesario para realizar las pruebas, por tanto para una implementación a tiempo real debería estudiarse si compensa utilizar un número

muy elevado de repeticiones o por el contrario es más rentable disminuir el rendimiento a costa de ganar en tiempo.

	10 iteraciones					50 iteraciones					100 iteraciones				
	A	E	I	O	U	A	E	I	O	U	A	E	I	O	U
A	192	8	0	0	0	194	6	0	0	0	195	5	0	0	0
E	0	117	81	2	0	0	134	65	1	0	0	129	69	2	0
I	2	75	121	1	1	0	72	127	0	1	1	67	131	1	0
O	1	1	0	161	37	1	1	0	173	25	1	1	0	173	25
U	0	0	0	50	150	0	0	0	30	170	0	0	0	26	174

Tabla F.1: Matrices de confusión obtenidas con AdaBoost + J4.8, utilizando 10, 50 y 100 iteraciones. La primera da un acierto medio del 74,1 %, la segunda 79,8 % y la tercera 80,2 %.

F.1.2. Fusión de clases

Visto que la mayoría de los fallos del clasificador se producen al confundir entre sí las clases *E-I* y *O-U*, en esta sección estudiaremos cuál sería el rendimiento si se uniesen.

F.1.2.1. Fusión de las clases E-I

En la tabla F.2 se muestran las matrices de confusión obtenidas al juntar en una las clases *E-I*. Para ello, se tomaron 200 muestras de esta nueva clase (100 escogidas al azar de cada letra), con el objetivo de que hubiese el mismo número de ejemplos de cada grupo.

	10 iteraciones				50 iteraciones			
	A	E-I	O	U	A	E-I	O	U
A	193	7	0	0	198	2	0	0
E-I	4	194	2	0	1	198	1	0
O	1	1	158	40	1	1	165	33
U	0	0	42	158	0	0	24	176

Tabla F.2: Matrices de confusión obtenidas con AdaBoost + J4.8, utilizando 10 y 50 iteraciones para la clasificación de las vocales fusionando las clases E-I. La primera da un acierto medio del 87,875 % y la segunda 92,125 %.

Como puede verse en los resultados, los porcentajes de acierto aumentan considerablemente tras esta fusión. Esto se debe a que la gesticulación normal para pronunciar esas dos letras puede ser muy similar, lo que hace que en ocasiones el clasificador sea incapaz de distinguirlas.

F.1.2.2. Fusión de las clases O-U

Siguiendo en la misma línea se quiere dar una vuelta de tuerca más y ver hasta qué porcentaje de acierto podría obtenerse juntando las clases *O-U* que, aunque en menor medida, también se confunden entre sí.

	10 iteraciones			50 iteraciones		
	A	E-I	O-U	A	E-I	O-U
A	196	4	0	196	4	0
E-I	2	197	1	1	199	0
O-U	0	1	199	0	1	199

Tabla F.3: Matrices de confusión obtenidas con AdaBoost + J4.8, utilizando 10 y 50 iteraciones para la clasificación de las vocales fusionando las clases E-I y O-U. La primera da un acierto medio del 98,67 % y la segunda 99 %.

El resultado es, como puede verse en la tabla F.3, de casi un 100 % de acierto, lo que deja claro que estas tres clases son muy distinguibles entre sí.

F.1.3. Clasificaciones por canal

En este apartado van a mostrarse los resultados de clasificaciones de las 5 vocales utilizando vectores de características obtenidos para cada canal de uno en uno, en contraposición con los que se muestran en las tablas del apartado F.1.1.3, que corresponden a la concatenación de todos estos vectores.

	10 iteraciones
	True Positives (%)
Canal 1	59,2
Canal 2	55,1
Canal 3	58,8
Canal 4	63,8
Canal 5	57,1
Canal 6	53,2
Canal 7	55,9
Canal 8	39,7

Tabla F.4: Porcentajes de acierto calculados por el clasificador al proporcionarle como características los vectores obtenidos para cada canal por separado.

En la tabla F.4 puede comprobarse que ninguno de los canales por separado alcanza el porcentaje de acierto que ofrece la concatenación de todos ellos (74,1 % para el mismo número de iteraciones). El canal 8, además, proporciona un resultado especialmente bajo, lo que hace pensar que colocar los electrodos en esa localización no supone una buena

idea. El canal 4, situado en el músculo *Orbicularis Oris Superior* (entre el labio superior y la nariz), proporciona el mejor resultado al ser esa zona una de las que más se mueven al realizar las pronunciaciones. Por lo que respecta a los demás, parece algo normal que cada uno de los canales individualmente clasifique peor que la concatenación de todos ellos, ya que, cuanta más información útil se le ofrezca al clasificador, lo más probable es que genere mejores resultados.

F.2. Clasificaciones con recolocación de electrodos

Uno de los principales aspectos que se querían comprobar en este proyecto y que más dudas generaba era si la utilización de datos provenientes de una misma persona, pero adquiridos en distintos días afectaría a los resultados. Lo más lógico sería que el reposicionamiento de los sensores supusiese desplazamientos milimétricos en la localización, lo que provocaría que las señales electromiográficas grabadas, para una misma clase, pudiesen variar en mayor o menor medida.

Para comprobar el impacto de esta recolocación se diseñó una prueba utilizando muestras de vocales adquiridas en distintos días para una misma persona. Por un lado, se realizó una clasificación de 100 muestras de las 5 vocales adquiridas el mismo día, después se seleccionaron al azar 50 de esas muestras y se juntaron con otras 50 adquiridas en una sesión distinta. Así, con la utilización del mismo número de ejemplos en las dos pruebas se conseguiría no sesgar los resultados y conseguir unos porcentajes representativos.

	Misma sesión					Distintas sesiones				
	A	E	I	O	U	A	E	I	O	U
A	99	0	1	0	0	95	5	0	0	0
E	1	62	36	1	0	5	63	32	0	0
I	0	31	68	0	1	2	26	71	0	1
O	0	1	0	75	24	0	0	0	81	19
U	0	0	0	15	85	0	1	0	18	81

Tabla F.5: Matrices de confusión correspondientes a la clasificación con ejemplos obtenidos en una misma sesión y en dos sesiones distintas. La primera acierta en el 77,8 % de los casos, mientras que la segunda en el 78,2 %.

Como se puede observar en la tabla F.5 el porcentaje de acierto no ha bajado. Y no sólo eso, sino que incluso ha mejorado. La conclusión que puede extraerse es que el reposicionamiento de los electrodos, si se realiza siguiendo el protocolo adecuado y con precaución no tiene porqué afectar a los resultados. El hecho de que haya mejorado el porcentaje no es nada significativo, ya que es bastante normal que se produzcan variaciones al realizar pruebas con distintas muestras; puede verse en las clasificaciones realizadas con 200 muestras del primer experimento, donde se obtuvo un 74,1 %. Pero el comprobar que no se produce una caída drástica en el rendimiento es una gran noticia que permi-

tirá proseguir con investigaciones para la ampliación del vocabulario y la obtención de una amplia base de datos con muestras de todas las clases.

F.3. Clasificaciones de sílabas

En esta sección se mostrarán más resultados de los esquemas de clasificación explicados en el capítulo 5 que, por falta de espacio, no se mostraron ahí. Las pruebas realizadas corresponden a ejecuciones con 10, 50 y 100 iteraciones.

F.3.1. Clasificador de 30 clases

Las figuras F.4, F.5 y F.6 son representaciones a color de las matrices de confusión correspondientes al clasificador multiclase de 30 clases para ejecuciones con 10, 50 y 100 iteraciones, respectivamente. Se puede ver que, conforme se aumenta el número de iteraciones, el color de la diagonal principal se acerca más a los tonos cálidos, mientras que las diagonales secundarias que aparecen cambian su color a tonos azules más oscuros. Estas diagonales secundarias obedecen a las confusiones que se producen en el clasificador por sílabas con las mismas terminaciones.

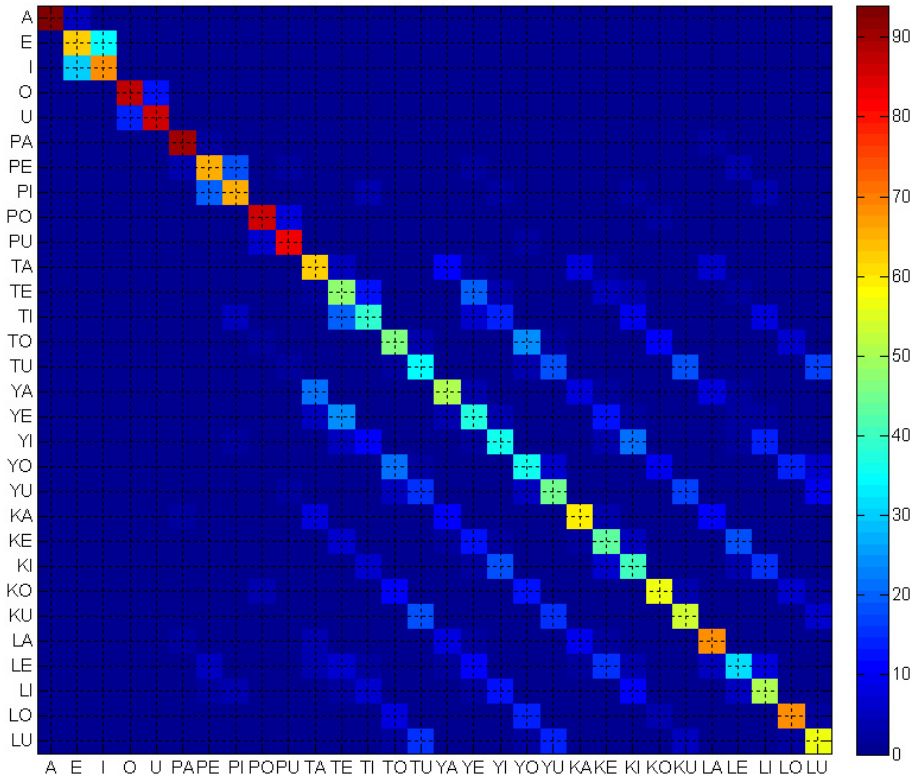


Figura F.4: Matriz de confusión clasificador multiclase con 10 iteraciones del Boosting.

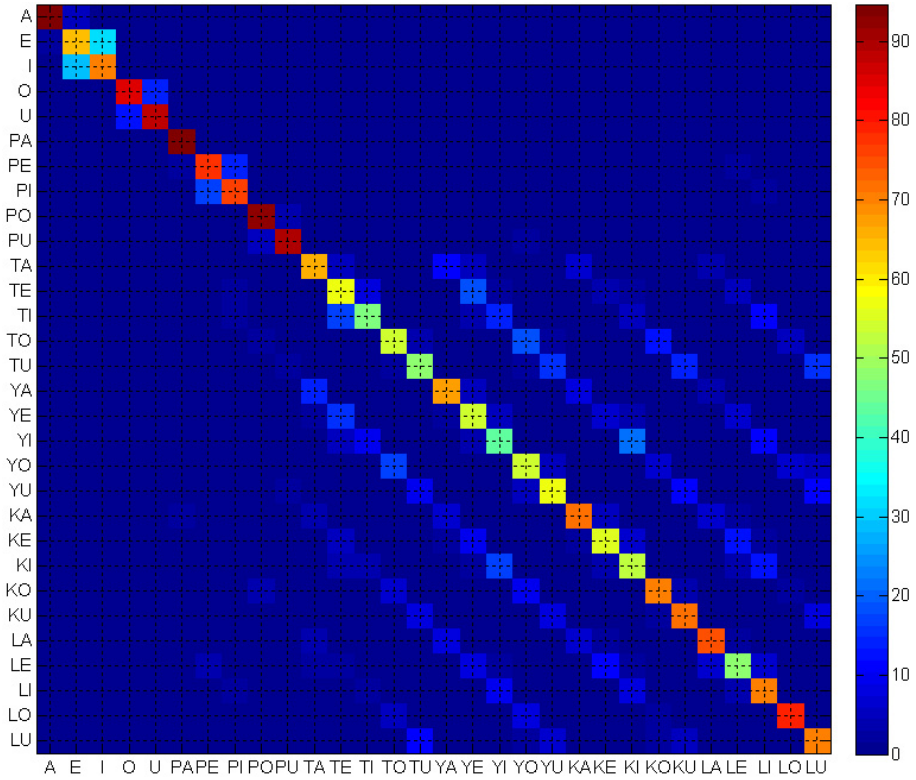


Figura F.5: Matriz de confusión clasificador multiclase con 50 iteraciones del Boosting.

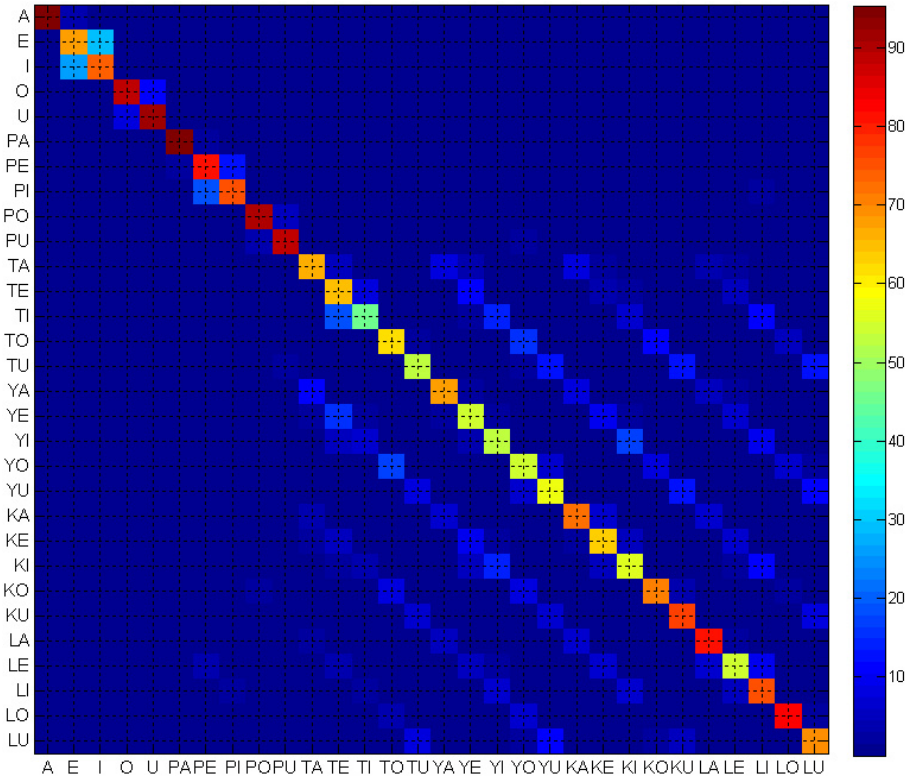


Figura F.6: Matriz de confusión clasificador multiclase con 100 iteraciones del Boosting.

En la figura F.7 se muestra una comparativa de los porcentajes de *true positives* correspondientes a cada una de las clases según el número de iteraciones. La media de acierto para 10 iteraciones es del 58,33 %, para 50 iteraciones del 68,16 % y para 100 iteraciones, como ya se vio en el apartado 5.1.1, del 70,93 %.

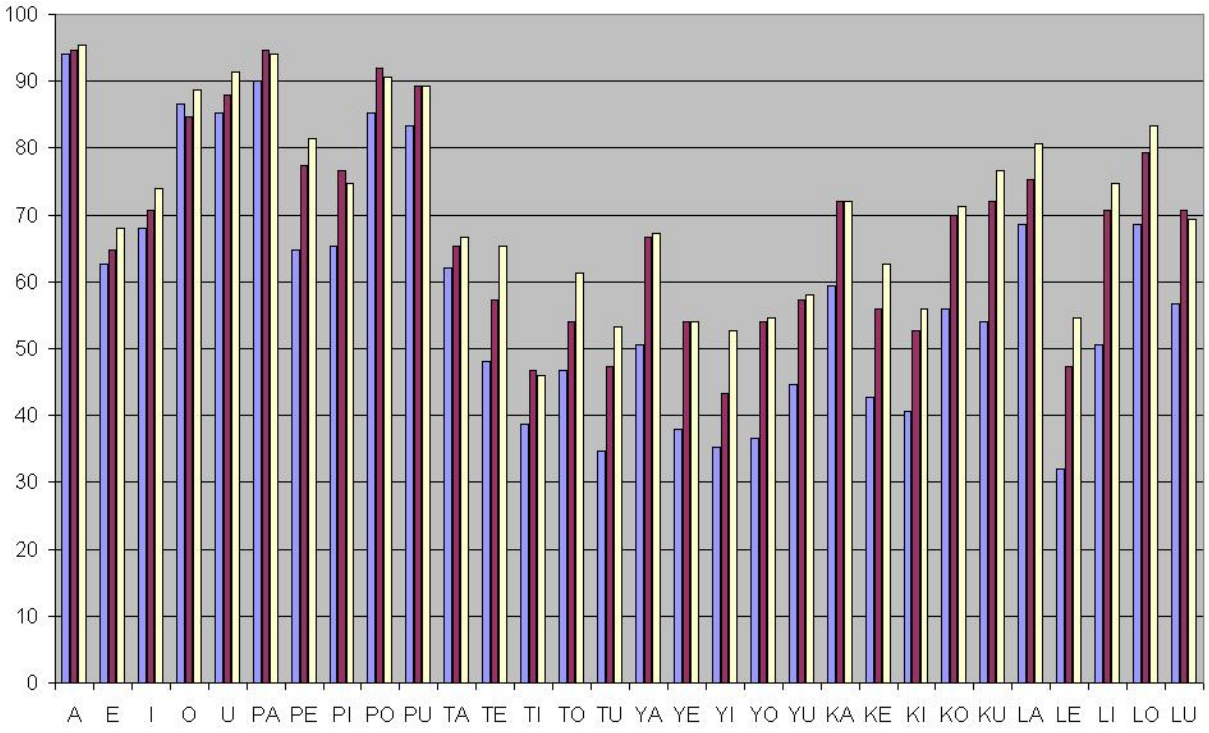


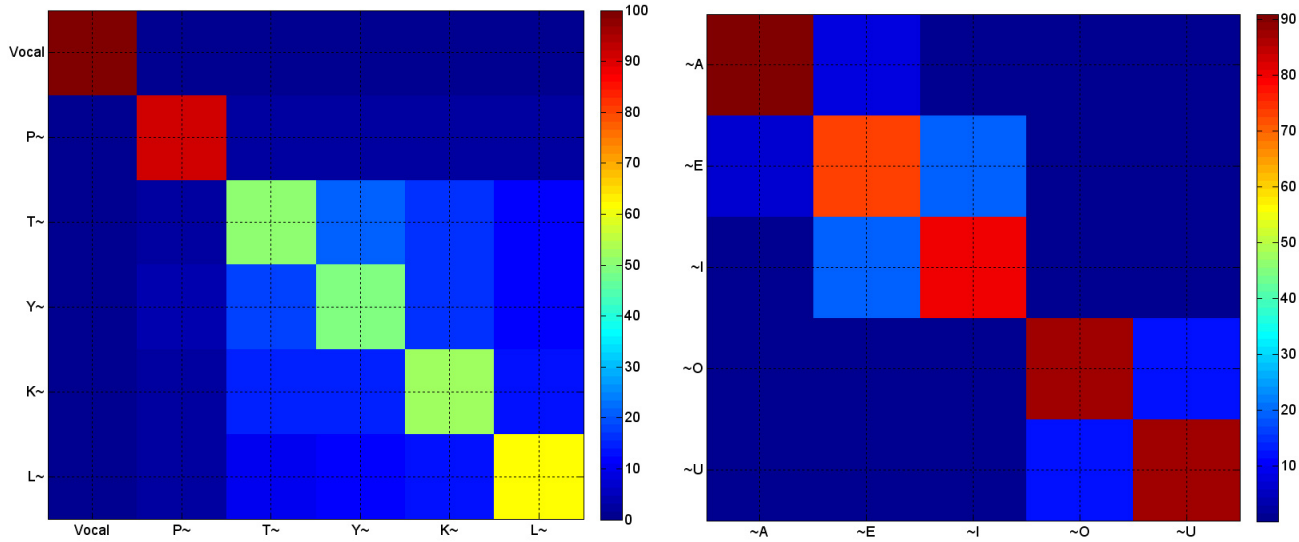
Figura F.7: Comparativa en el porcentaje de *true positives* conseguidos para el clasificador multiclase con 10, 50 y 100 iteraciones.

F.3.2. Clasificador matricial

En las matrices de confusión representadas en las figuras F.8, F.9 y F.10 se contempla la mejora experimentada en los resultados al incrementar el número de iteraciones ejecutadas en el *Boosting* al ir cambiando los colores de la diagonal principal hacia los tonos más rojos. Como puede observarse, el clasificador que actúa distinguiendo las sílabas según su terminación da unos resultados mucho mas altos en media que el encargado de distinguir los comienzos. Así que una buena forma de mejorar este esquema sería encontrar algún clasificador que distinguiese mejor esos patrones, en lugar de utilizar el actual. Esto requeriría muchas pruebas para determinar qué máquina de aprendizaje proporciona un mayor incremento en los porcentajes de reconocimiento.

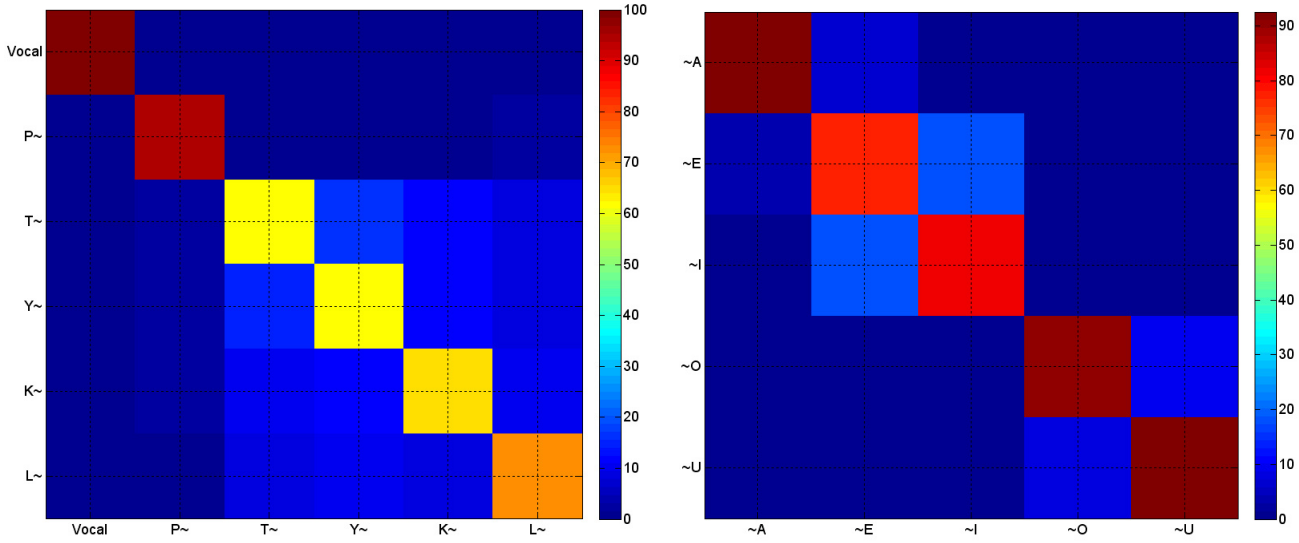
En el gráfico de barras de la figura F.11 se muestra la comparativa en porcentaje de los aciertos que consigue cada clasificador según el número de iteraciones que se realicen. Para 10 iteraciones se obtiene, de media, un 56,69 % de acierto; para 50 se alcanza el 65,83 % y para 100 iteraciones reconoce correctamente el 67,65 % de las muestras.

En todos los casos este esquema proporciona peores resultados que el multiclase estándar en cuanto a porcentajes de acierto. Sin embargo, si se ejecutase en dos procesadores distintos, como se propuso en las conclusiones del apartado 5.1.4, tardaría menos de la mitad del tiempo en entrenarse y clasificar, lo que podría ser suficiente razón para decantarse por este modelo en lugar del anterior.



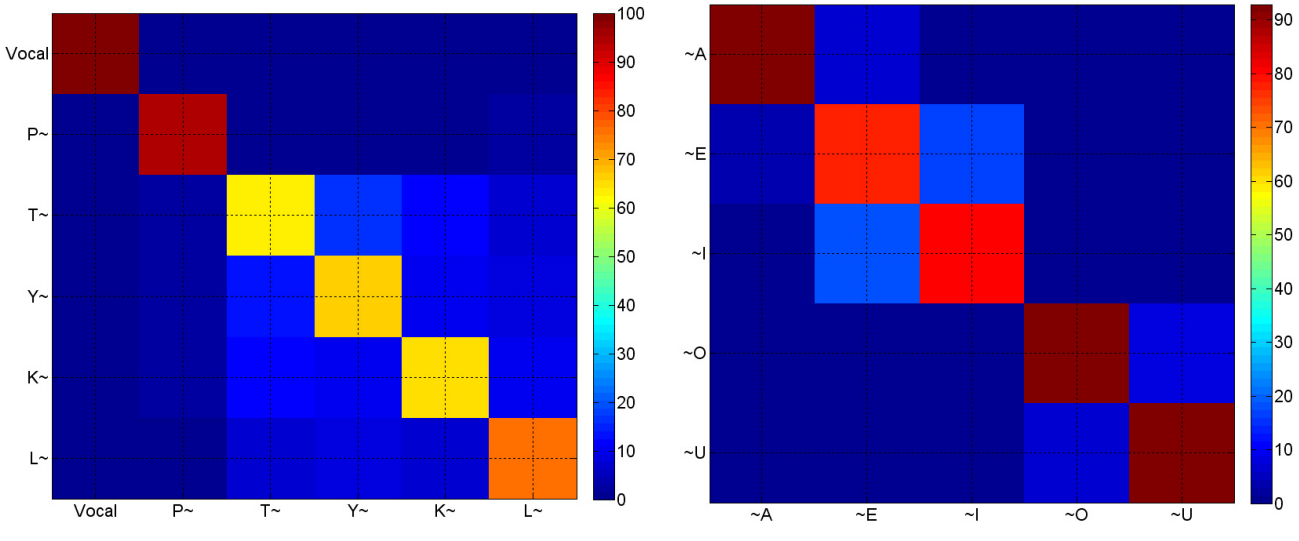
(a) (b)

Figura F.8: Matrices de confusión clasificador matricial con 10 iteraciones del Boosting.



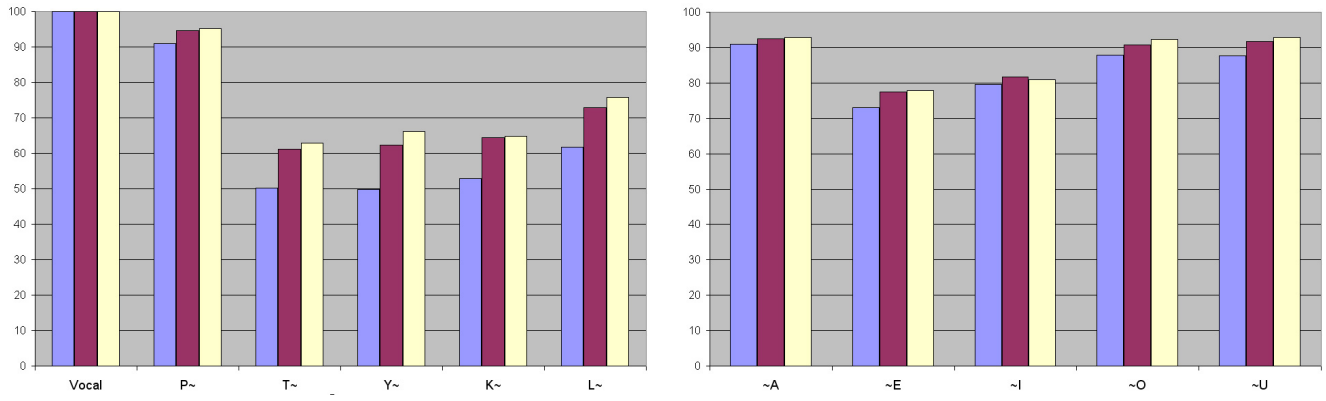
(a) (b)

Figura F.9: Matrices de confusión clasificador matricial con 50 iteraciones del Boosting.



(a) (b)

Figura F.10: Matrices de confusión clasificador matricial con 100 iteraciones del Boosting.



(a) (b)

Figura F.11: Comparativa en el porcentaje de *true positives* conseguidos para los dos clasificadores que forman el esquema matricial con 10, 50 y 100 iteraciones.

F.3.3. Clasificadores condicionales

En este apartado se muestran a color las matrices de confusión correspondientes a los clasificadores condicionales ejecutados sólo para 100 iteraciones. También se muestran, mediante diagramas de barras, comparativas que muestran las mejoras en los resultados medios de cada clasificador, en lugar de las mejoras producidas en cada clase individual.

F.3.3.1. Clasificador Fila-Columna Condicional

La figura F.12 representa las matrices de confusión correspondientes a cada uno de los 6 clasificadores condicionales que actúan según el comienzo determinado por el clasificador fila explicado anteriormente. Estas matrices corresponden a la ejecución de AdaBoost + J4.8 ejecutado 100 veces. Como se ve, el clasificador que menos aciertos consigue es el de las vocales, estando todos los demás próximos a una media del 90 % de reconocimiento.

La gráfica de la figura F.13 es una comparación, cambiando el número de iteraciones entre 10, 50 y 100, de los porcentajes medios de acierto obtenidos por cada uno de los 6 clasificadores encargados de distinguir las terminaciones de una sílaba según el comienzo determinado por el clasificador fila. Es importante no confundir los datos de esta gráfica con la del apartado F.3.2, ya que pese a ser similares, ésa lo que muestra es el porcentaje de *true positives* obtenidos para cada una de las clases y ésta el porcentaje medio de acierto de cada clasificador.

Las probabilidades medias conseguidas por estos clasificadores son: para 10 iteraciones, 85,52 %; para 50, 87,87 % y para 100, 88,94 %. Esto debe multiplicarse por el resultado que consigue el clasificador fila correspondiente al mismo número de iteraciones y se consiguen para este esquema unos porcentajes de acierto de 57,88 %, 66,64 % y 68,89 %, dependiendo del número de veces que se repita el *Boosting*.

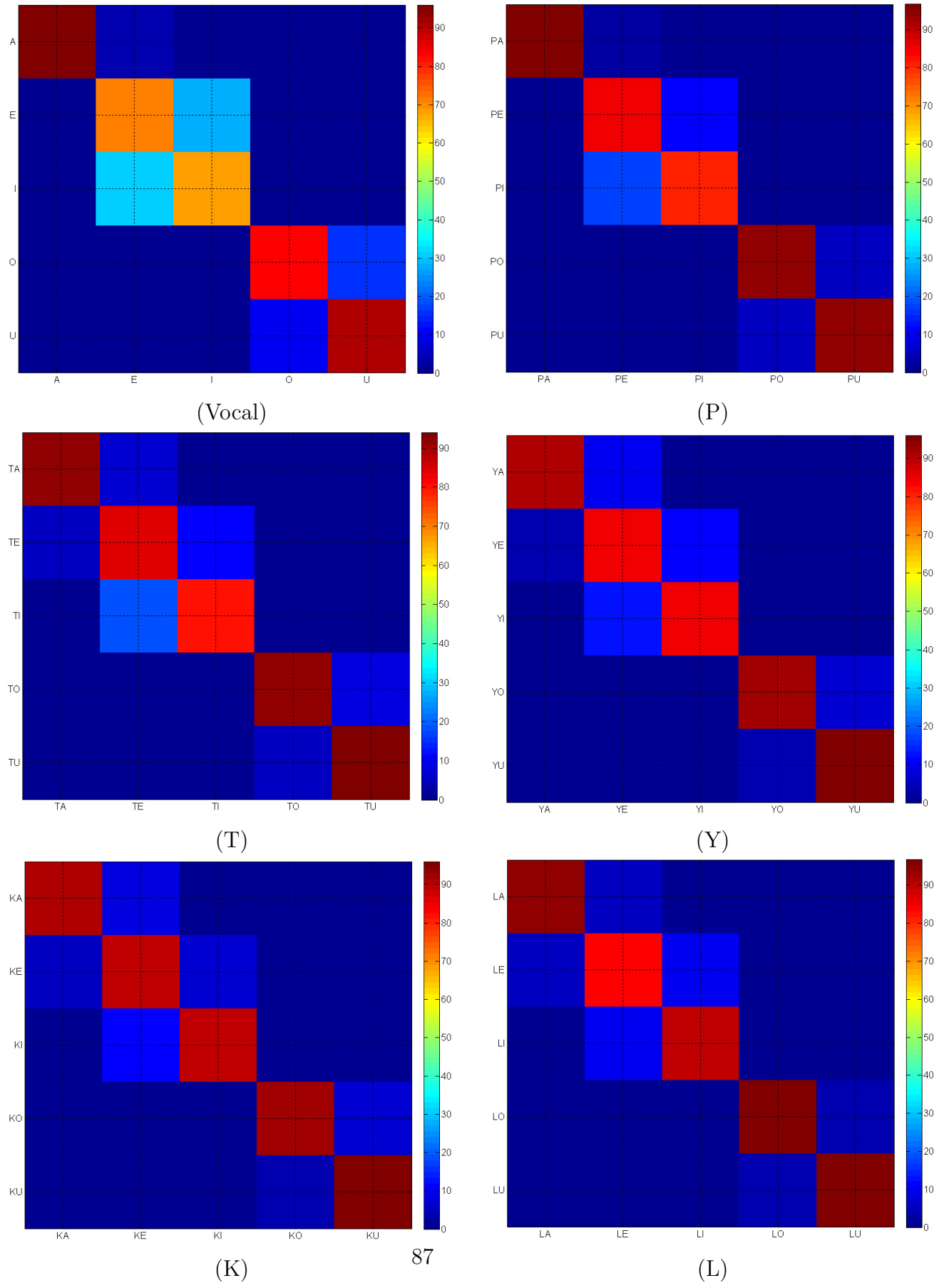


Figura F.12: Matrices de confusión correspondientes a los clasificadores condicionales por terminación con 100 iteraciones.

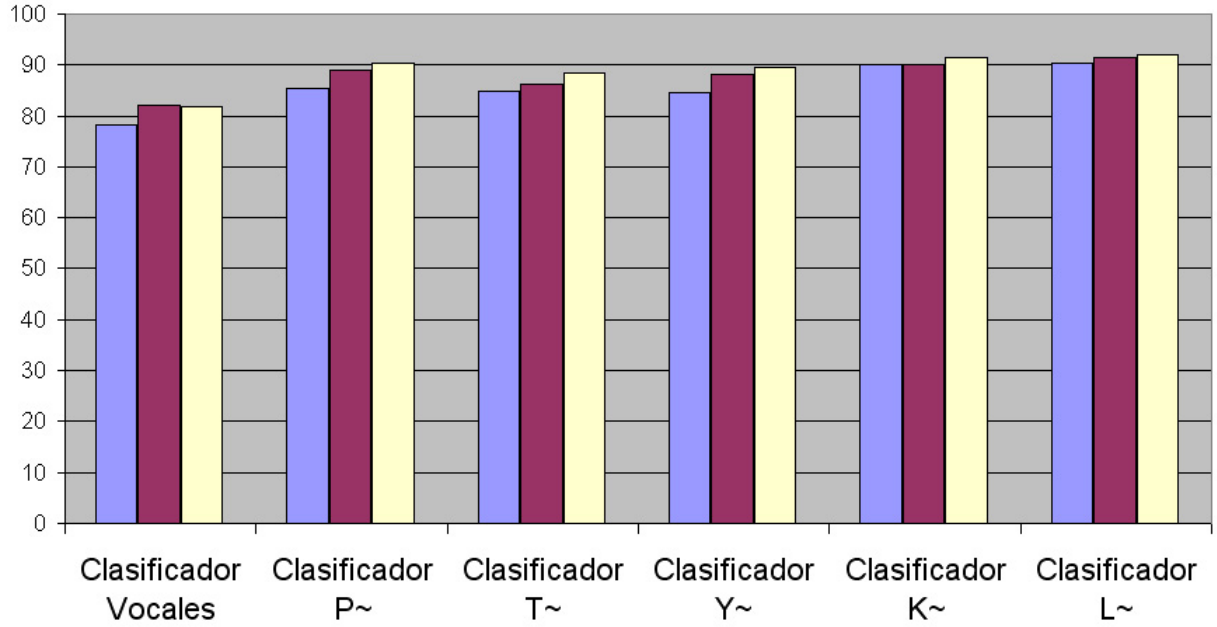
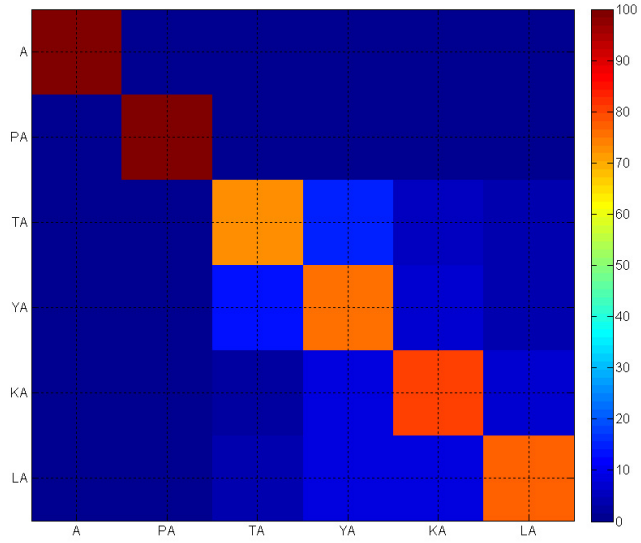


Figura F.13: Comparativa en el porcentaje de *true positives* conseguidos para los clasificadores condicionales de terminación con 10, 50 y 100 iteraciones.

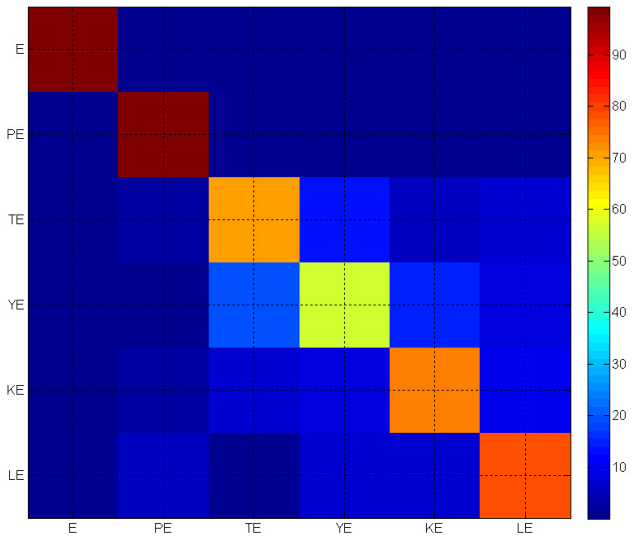
F.3.3.2. Clasificador Columna-Fila Condicional

La representación de las matrices de confusión a color de los 5 clasificadores diseñados para determinar el comienzo de una sílaba iterando 100 veces pueden verse en la figura F.14. El aspecto que presentan todas ellas es muy similar al que tenía la correspondiente al clasificador matricial de filas: las vocales y las sílabas que comienzan en P se diferencian muy bien (color muy rojo en la matriz), mientras que el resto se confunden en mayor medida entre ellas.

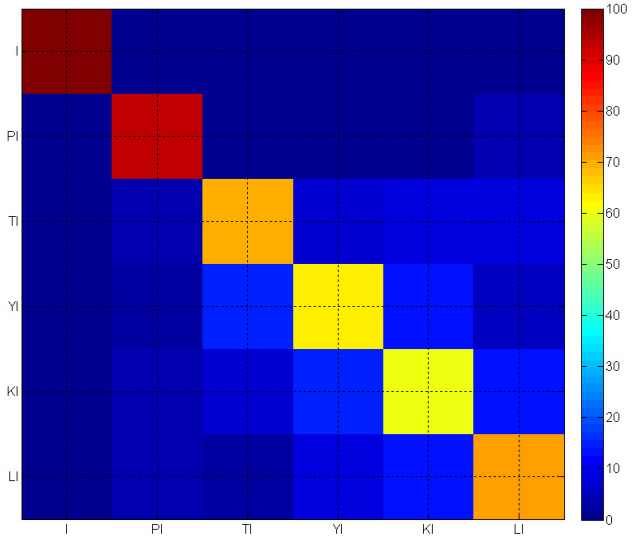
Al igual que en el subapartado anterior, en la gráfica F.15 se comparan los porcentajes de acierto de cada uno de estos clasificadores según se ejecute el árbol de decisión 10, 50 o 100 veces. Se ve como el clasificador de sílabas que terminan por A consigue unos mejores resultados, pero los todos ellos son bastante estables, estando próximos a la media total que, en el caso de 10 iteraciones es del 70,83 % de acierto; con 50 iteraciones 78,49 % y para 100: 79,56 %. Al multiplicar estas probabilidades por las que proporcionaba el clasificador de terminaciones correspondiente al mismo número de iteraciones en cada caso, se obtienen unas medias para este esquema de 59,4 %, 68,13 % y 69,48 % para las 10, 50 y 100 repeticiones.



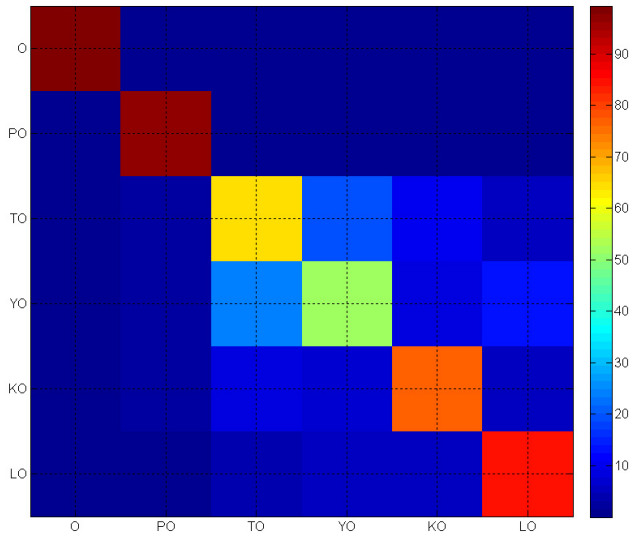
(A)



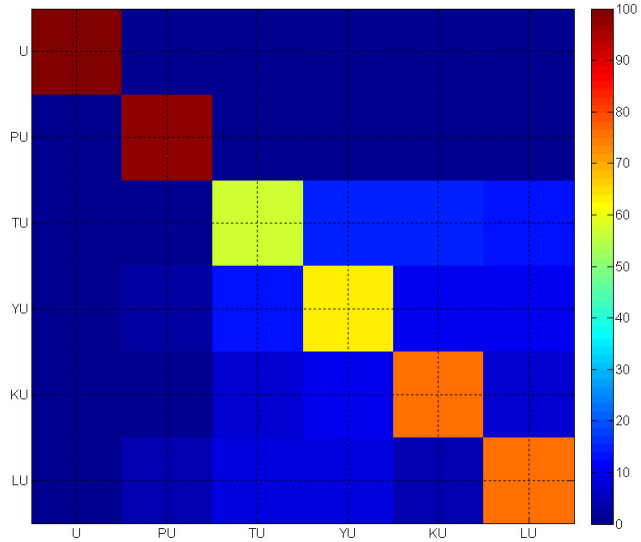
(E)



(I)



(O)



(U)

Figura F.14: Matrices de confusión correspondientes a los clasificadores condicionales por comienzo 100 con iteraciones

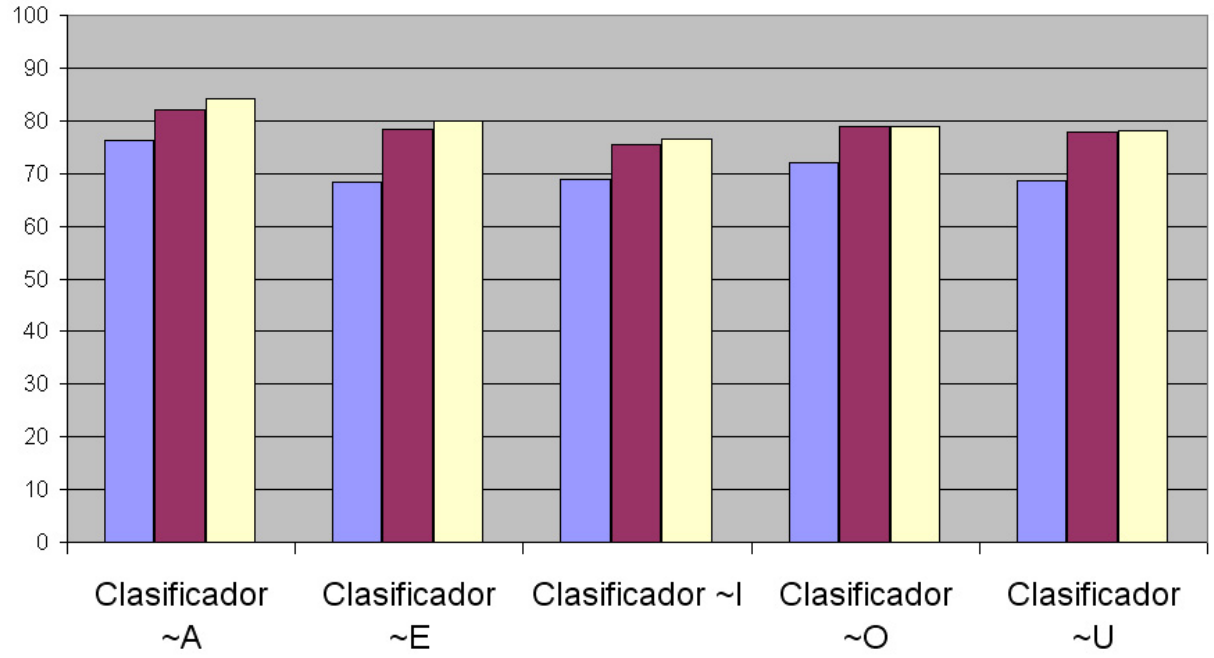


Figura F.15: Comparativa en el porcentaje de *true positives* conseguidos para los clasificadores condicionales de comienzo con 10, 50 y 100 iteraciones.

F.4. Clasificaciones vocal-sílaba

Otro esquema de clasificación diseñado ha sido uno que se encargue de distinguir entre sílabas vocálicas y consonánticas, sin aportar más información que ésa. Podría utilizarse como clasificador de apoyo, ejecutándose en paralelo, para sílabas en las que se producen bastantes confusiones y aseguraría, en gran medida, si una sílaba dada corresponde a una vocal o no.

Para la realización de estos tests se emplearon las mismas muestras que para el resto de clasificaciones de las 30 sílabas, por tanto la clase de las sílabas vocálicas estaba compuesta por 750 ejemplos, mientras que había 3750 ejemplos de sílabas consonánticas. Esto no suele ser recomendable por el sesgo que puede producir la diferencia en el número de datos de las distintas clases, sin embargo, para este caso particular y dada la naturaleza de la distinción que se quería hacer, se hizo una excepción.

La tabla F.6 muestra las matrices de confusión obtenidas para 10 y 50 iteraciones. Los resultados, como puede observarse son de un acierto cercano al 100 % (99,78 y 99,82 % respectivamente).

	10 iteraciones		50 iteraciones	
	S. Vocálica	S. Consonántica	S. Vocálica	S. Consonántica
S. Vocálica	3745	5	3746	4
S. Consonántica	5	745	4	746

Tabla F.6

F.5. Clasificaciones sílaba-no sílaba

El último esquema que se va a mostrar en este anexo es el sistema de clasificación diseñado para distinguir entre señales pertenecientes a lo que es una sílaba de lo que no lo es. Esto puede ser esencial cuando quiera realizarse una aplicación en tiempo real, ya que deberá existir un mecanismo que pueda diferenciar una señal correspondiente a una pronunciación del resto de movimientos faciales.

Para este caso sí se han utilizado el mismo numero de muestras para las dos clases. Para la clase *sílaba* se tomaron los 150 ejemplos disponibles para cada uno de los 30 grupos y, por otro lado, se seleccionaron 4500 muestras de señal correspondientes a momentos donde no existía pronunciación, con una duración idéntica a los ejemplos mencionados y se les extrajeron las mismas características. Con esto se entrenó la máquina de clasificación ya mencionada (AdaBoost + árbol J4.8) y se probó a iterar 10 y 50 veces. Los resultados se muestran en la tabla F.7.

	10 iteraciones		50 iteraciones	
	Sílaba	No-Sílaba	Sílaba	No-Sílaba
Sílaba	4496	4	4497	3
No-Sílaba	3	4497	2	4498

Tabla F.7

El porcentaje de acierto obtenido es del 99,92 % y 99,94 % para 10 y 50 iteraciones, respectivamente. Con esto queda claro que la diferenciación entre sílabas y señales que no corresponden a ellas es algo sencillo de conseguir.