Trabajo de Fin de Grado

Integración de reconocimiento de voz con el análisis del discurso.

Voice recognition integration with discourse analysis.

*H*aniel Lorenzo Martín Arteaga

La Laguna, 7 de Junio de 2016

D. **Francisco Javier Martínez García**, con N.I.F. 12.345.678-X profesor Titular de Universidad adscrito al Estadística, Investigación Operativa y Computación del Departamento de la Universidad de La Laguna, como tutor

D. **Julio Brito Santana**, con N.I.F. 12.345.678-X profesor Titular de Universidad adscrito al Departamento de Estadística, Investigación Operativa y Computación de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Integración de reconocimiento de voz con el análisis del discurso."

ha sido realizada bajo su dirección por D. **Haniel Lorenzo Martín Arteaga** N.I.F. 78.855.096-X.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de junio de 2016.

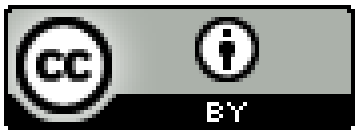
Agradecimientos

Me gustaría agradecer a mis profesores, por su labor y su ayuda en la gestión y durante el desarrollo de este trabajo.

A mi familia y amigos, por compartir conmigo esta etapa de mi vida.

Pero sobre todo, gracias a mi padre por su ayuda incondicional.

Licencia



Resumen

El presente Trabajo Fin de Grado analiza la posibilidad de añadir reconocimiento de voz como entrada dentro de un proyecto mucho más ambicioso que contempla el Análisis Automatizado del Discurso.

El Análisis del Discurso es una disciplina que estudia la riqueza y el significado del lenguaje utilizado dentro de un corpus (novela, discurso político, entrevista, etc) en su contexto, utilizando un conjunto de herramientas que permiten entre otros, comparar con otros corpus.

Un equipo multidisciplinar de investigadores (lingüistas e informáticos) han desarrollado una conjunto de herramientas denominada SISAD (Sistema Informático de Soporte al Análisis del Discurso) que permite el tratamiento semiautomático de los corpus.

SISAD ya cuenta con más de 10 años de uso y desarrollo y facilita a usuarios lingüistas o no el análisis de discursos ordinarios o literarios. Organiza y representa el conocimiento en formato textual para su procesamiento. Recupera además la información y extracción del conocimiento en un corpus dado, utilizando procedimientos estadísticos y la experiencia del analista recogida en un árbol semántico.

Dentro de sus aplicaciones ha permitido estudiar el discurso literario de la novela franco-antillana, discursos ordinarios recogidos en poblaciones particulares para estudiar realidades socioculturales y la identidad cultural en el contexto de relaciones personales o grupales.

Después de estos diez años de uso se ha replanteado la mejora del sistema para que sea más automática y proveerla de las últimas técnicas en aprendizaje y lematizado automático además de proveer una interfaz

mejorada y una arquitectura cliente – servidor que permita a los investigadores trabajar desde cualquier lugar utilizando Internet.

Además de los enormes retos en inteligencia artificial y sistemas de almacenamiento big-data se ha planteado la posibilidad de introducir textos directamente del audio de las entrevistas objeto de varios estudios sobre la vertiente francófona de la Guayana francesa.

Las posibilidades de este sistema de análisis del discurso son muy amplias e incluyen análisis del discurso de orientación política, mediático y de redes sociales. Comparativas de discurso oral frente al escrito. Avances en la enseñanza-aprendizaje de lenguas, adquisición automática de información léxica y sistemas avanzados de traducción automática.

Con estos antecedentes se considera fundamental y de ahí la propuesta de trabajo de fin de grado, contenida en este documento, de comprender la física que existe detrás de las ondas que conforman la voz, el conocimiento de los sistemas de programación en los que se traducen los diferentes algoritmos y el estudio el estado del arte y las posibilidades de incluir un módulo de reconocimiento de voz a la nueva versión de SISAD actualmente en desarrollo.

En este TFG se incluye tanto el estudio y análisis teórico del reconocimiento del habla, como diferentes posibilidades prácticas con software libre y el código asociado para que funcionen estas opciones.

Palabras clave: Reconocimiento de voz, corpus, modelos de lenguaje, patrones acústicos, lematizador, estado del arte, aplicación, análisis, oportunidades.

Abstract

This Final Project analyze the possibility of adding text recognition as entrance into a much more ambitious project that includes the Automated Analysis of the Discourse.

Discourse Analysis is a discipline that studies the richness and meaning of the language used within a corpus (novel, political speech, interview, etc.) in context, using a set of tools that allow us to compare it with other corpus.

A multidisciplinary team of researchers (including linguists and computer scientists) have developed a toolkit called SISAD (computer support system discourse analysis) that allows the semiautomatic treatment of corpus.

SISAD already has more than 10 years of use and development and facilitates to linguists users or not the analysis of ordinary or literary discourses. Organizes and represents knowledge in textual format for its processing. It also retrieves information and extraction of knowledge in a corpus given according to statistical procedures and the analyst's experience collected in a semantic tree corpus.

Among its applications it has allowed us to study the literary discourse of Franco-Antillean novel, ordinary speeches collected in particular populations to study socio-cultural realities and cultural identity in the context of personal or group relationships.

After ten years of use it has been redesigned to improve the system to make it more automatic and provide it with the latest techniques in learning and automatic lemmatized, it also provides an improved interface and a client – server architecture that allows researchers to work from anywhere in the planet using Internet.

In addition to the enormous challenges in artificial intelligence and big –data storage systems, it has raised the possibility of entering text directly from audio interviews subject of several studies on the francophone side of French Guiana.

The possibilities of this analysis system of the discourse are very broad and include discourse analysis of political, media and social networking orientation. Comparisons of oral speech to writing. Advances in teaching and learning of languages, automatic acquisition of lexical information and advanced automatic translation systems.

With this background it is consider fundamental and that is the reason of this final project, contained herein, to understand the physics that exist behind the waves that make up the voice, the knowledge of programming systems in which different algorithms are translated and the study the state of the art and the possibilities of including a voice recognition module to the new version of SISAD currently under development.

In this Final Project it has included both, the study and theoretical analysis of speech recognition, as different practical possibilities with free software and the associated code to make these options work.

Keywords: *Speech Recognition, corpus, language models, acoustic patterns, stemmer, state of the art, application, analysis, opportunities.*

Índice General

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

Los sistemas de reconocimiento automático del habla (SRAH) han pasado de las películas de ciencia ficción como "2001: Una Odisea del Espacio" de 1968 a estar en nuestros smartphones como el programa Siri de Apple o Google VoiceSearch para plataformas Android. Lo que inicialmente pudo haber sido una curiosidad científica ahora es una realidad y una fuente de negocio. Existen en la actualidad SRAH comercializados y con excelente rendimiento, ya no nos sorprende que al hacer una llamada telefónica a nuestro banco, proveedor de servicios de telefonía o seguro médico nos responda una máquina; en ocasiones finalizamos la gestión de forma satisfactoria sin que en ningún momento hayamos hablado o interaccionado con otra persona. Los SRAH están y se quedarán, ya están implementados en domótica, ayudas a discapacitados o traducción automática del habla. Las posibilidades son infinitas y sólo nuestra imaginación les pone límites, no nos extraña que ya en los años setenta fuera el sistema de defensa de los EEUU quien patrocinara los estudios iníciales a través de la agencia de proyectos DARPA (Defense Advanced Research Project Agency).

En la actualidad se están desarrollando tanto sistemas orientados hacia aplicaciones específicas como los comentados Siri o Google VoiceSearch como sistemas de reconocimiento del habla en condiciones naturales, independientes del operador, del ruido e incluso el idioma. La tecnología se ha transformado tanto que en unos pocos años el sistema anterior queda obsoleto si no se actualiza permanentemente. Nosotros hemos decidido hacer una valoración crítica de los sistemas actuales, para esa valoración nos basaremos en las técnicas de análisis DAFO. Nuestro trabajo se ha limitado a los sistemas cuyos autores hayan liberado para su uso gratuito y disponible a través de internet.

Nuestro trabajo forma parte de un proyecto más ambicioso de desarrollo de una herramienta concebida "para facilitar a usuarios lingüistas o de otras áreas el trabajo de análisis de discurso ordinarios o literarios" (Cruz y Brito, 2014) llamada Sistema Informático de Análisis del Discurso. El objetivo final no es una mera transcripción del habla a texto sino que el sistema ayude tanto a la comprensión de contenidos como a la comparación entre discursos.

Los objetivos que nos hemos planteado son los siguientes:

- Resumir el estado del arte actual en las tecnologías de los SRAH.
- Describir las tecnologías de software libre, bajo licencias no comerciales.
- Usar las tecnologías disponibles y valorarlas críticamente según técnicas DAFO.
- Diseñar nuestros propios algoritmos de entrada de datos pero evitar la variabilidad que pueda generar si usamos algoritmos diferentes.

A la hora de presentar este trabajo vamos a hacer una breve descripción del sonido como fenómeno físico, repasaremos de forma somera la fisiología de la audición, comentaremos algunos conceptos de lingüística, describiremos en qué consiste una técnica DAFO y finalmente describiremos el estado del arte en sistemas de reconocimiento automático del habla.

A continuación presentaremos un capítulo de material y métodos donde vamos a presentar las características técnicas de los programas de RAH que hemos usado y nuestro método de comparación. En el capítulo final haremos unos comentarios generales a modo de conclusión.

Capítulo 2.
El sonido y el habla

2.1 Aspectos físicos del sonido

El sonido es una onda mecánica que se genera debido a la vibración de moléculas de aire y se propaga a diferente velocidad según las características del medio.

Características físicas de un sonido son: la intensidad (que hace referencia a su energía y se mide en dB), el tono (que hace referencia a la frecuencia y se mide en Hz) y el timbre (que hace referencia a la complejidad de la onda sonora). Un sonido que esté formado por una onda sinusoidal pura es inexistente en la naturaleza, el sonido es siempre una onda compleja que resulta de la suma de ondas sinusoidales de diferente amplitud, fase y frecuencia. Existe una herramienta matemática, llamada transformación de Fourier que permite extraer de una onda compleja sus componentes sinusoidales, lo que facilita su almacenamiento digital.

El sonido se puede almacenar usando un transductor llamado micrófono que transforma la señal mecánica en eléctrica. Los micrófonos pueden ser de varios tipos como de carbón o de bovina móvil, casi en desuso, o los más modernos de cinta, piezoeléctricos o capacitivos. La señal eléctrica resultante se almacena en un soporte analógico o digital. En la actualidad se usan casi exclusivamente los digitales, por sus superiores ventajas en todos los campos de aplicación. Una información digital se pude incorporar a un algoritmo y sufrir modificaciones matemáticas que permiten su uso en cualquier campo de la lingüística: reconocimiento de la voz, análisis sintácticos, análisis morfológicos, desarrollos de gramáticas, conversión de texto en habla.

2.2 Fisiología del sonido

Los cambios de presión del aire que genera la onda de sonido llegan al oído y sufre una serie de transformaciones hasta que llega a la corteza cerebral y se procesa, siendo conscientes de la información recibida. En la primera fase, el oído externo y medio actúan como amplificadores, de manera que la energía que porta la onda de presión se transmita a la cóclea del oído interno. El oído interno actúa como un transductor, por medio de procesos bioquímicos descompone la señal auditiva en sus componentes sinusoidales, de las cuales recogen la amplitud y fase, las células ciliadas –verdadero transductor- las codifican a una señal eléctrica que se incorpora al nervio auditivo. Los centros de proceso serían el núcleo coclear que procesa y reenvía información al complejo olivar superior y al núcleo colículo inferior y desde este al tálamo y corteza, donde recibe un último procesamiento. La señal se procesa según un nivel jerárquico de complejidad, así, el complejo olivar superior permite localizar el origen espacial del sonido, en el colículo inferior la señal interactúe con el movimiento (útil para evitar situaciones de peligro) y en el tálamo y la corteza recibe el procesamiento más complejo y superior, que es especialmente relevante para percibir las sensaciones musicales y la comprender la palabra (Fitzpatrick, D. 2007).

Los seres humanos pueden detectar sonidos con un rango de frecuencias de entre 20 Hz y 20 kHz. El umbral auditivo a la presión sonora es de una billonésima de watt por metro cuadrado, por el contrario niveles superiores a 100 dB (en el rango del miliwatt por metro cuadrado) son peligrosos pues dañan el oído y además se perciben como dolor.

2.3 Concepto de lingüística aplicada a los SRAH

2.3.1 Lenguaje, lenguaje, idioma y habla

Definimos lenguaje la facultad o capacidad humana para comunicarse mediante sonidos articulados o códigos, no es sinónimo de lengua que es un sistema de signos orales o escritos además de sus reglas de combinación (Fernández. 2016). En etología o cibernética, por similitud, se usa para designar códigos de comunicación, entre animales o entre máquinas, en este sentido la academia de la lengua ha aceptado dentro del campo de la

informática la acepción de lenguaje como "Conjunto de signos y reglas que permite la comunicación con un ordenador" (RAE. 2015) Por eso es correcto decir la lengua o lenguas habladas en España pero no los lenguajes hablados en España. Sí es correcto decir lenguaje C, Java, Pascal o Cobol porque son códigos diferentes. Idioma puede ser equivalente a lengua cuando se habla de las lenguas nacionales modernas. El habla es la utilización que cada individuo hace de su lengua; el habla es individual y la lengua social. El dialecto es una variante de lengua o idioma que se habla en un determinado lugar o grupo social. La jerga sería la variante del lenguaje propia de una profesión (Fernández. 2016).

Al estudiar el lenguaje se suelen considerar cuatro áreas de estudio, que tienen interés para el desarrollo de sistemas expertos, son fonética, morfología, sintaxis y semántica. La fonética se encarga del estudio de "los sonidos del habla" (RAE), la morfología estudia "la estructura de las palabras y de sus elementos constitutivos" (RAE), la sintaxis "el modo en que se combinan las palabras y los grupos que estas forman para expresar significados, así como las relaciones que se establecen entre todas esas unidades" (RAE) y la semántica "el significado de las unidades lingüísticas y sus combinaciones (RAE).

2.3.2 Concepto de fonema, fono y alófono

El fonema es más un concepto abstracto que real, es la unidad mínima de lenguaje que se puede aislar, tanto sean vocales como consonantes. El español actual, al igual que otras muchas lenguas, no tiene una equivalencia entre fonema y grafema, sobre todo por razones históricas o etimológicas (RAE, 2010, p. 22). Así la j en jinete y la g giro suena igual –mismo fonema, representado como fonema /j/- pero tienen diferente escritura o grafía. Fono es cada uno de los segmentos de características acústicas particulares y con duración típica en que podemos dividir la secuencia sonora Los fonos similares, por convención, se representan por signos alfabéticos entre corchetes. En el apartado de física del sonido hemos visto como las ondas sonoras que los generan se caracterizan por un espectro de frecuencias. Es decir, la comunicación tiene lugar en un espectro o rango de frecuencias características, pero a su vez cada fono tiene su propio espectro de

frecuencias. El análisis espectrográfico permite descomponer las ondas sonoras en ondas más simples de frecuencias fijas.

El espectro del sonido sería la representación gráfica de las diferentes amplitudes de las frecuencias que conforman un sonido. El espectrograma es la representación gráfica de las frecuencias en función del tiempo (Fitzpatrick, D. 2007, pp. 340-341).

El número de fonos de una lengua está limitado, sin embargo en una misma palabra un fonema puede tener dos pronunciaciones ligeramente diferentes o la pronunciación de un fonema puede estar sujeta a variaciones locales e incluso individuales, pero los usuarios del idioma las percibirán así, como variantes, pero no como fonos diferentes. A esas variantes fonéticas es lo que llamamos alófonos. En otras palabras el fonema es una unidad abstracta y el fono una unidad práctica, del ejercicio individual de la lengua. Por ejemplo, Macperson (1975) citado Ríos Mestre (1999) distingue que el fonema /u/ puede sonar como semiconsonante cuando se sitúa entre consonante y vocal (bueno) como semivocal cuando está entre vocal y consonante (jaula) y como labio-alveolar entre vocales (ahuecar, este u otro) o en inicial de sílaba (hueco, las huertas. La mayoría de los autores aceptan que el español tiene 5 fonemas vocálicos y 19 consonánticos (Ríos Mestre,1999)

El interés principal de estos conceptos reside en que muestra lo complicado que es el funcionamiento de un sistema de reconocimiento de voz. Estos sistemas no son simplemente una interface que interpreta un fenómeno físico (onda sonora) y lo convierte en un código binario, entre otras características debe ser capaz de identificar los alófonos de la misma palabra. Además al hablar no tenemos un sistema de puntuación, el habla es dinámica y un SRAH debe distinguir palabras y frases.

2.3.3 Concepto de corpus

Entendemos el concepto de corpus como un texto que será objeto de estudio. Para el análisis del discurso un corpus es un texto que puede ser el capítulo de un libro, una entrevista o el conjunto de los textos que conforman un discurso político. Sin embargo el concepto de corpus se amplifica cuando se hace referencia a sistemas RSAH porque entonces se habla de enormes bases de datos que almacenan porciones de lenguaje (estos corpus). Muchos autores

han intentado definir el concepto de corpus, siendo una de las más citadas en la literatura es la del conocido lingüista John Sinclair (2005) que define corpus como "a collection of pieces of language text in electronic form, selected according to external criteria to represent, as far as possible, a language or language variety as a source of data for linguistic research." Todos los autores coinciden en hacer referencia a sus componentes principales, como: ser una colección de textos –base de datos-, estar estructurados según criterios lingüísticos y estar seleccionados según esos mismos criterios. Los criterios deben estar definidos claramente. Por ejemplo, si etiquetamos a una palabra como adverbio ya sabemos que modifica al verbo, por lo tanto el algoritmo caso de dos alternativas en las que traduzca una palabra como sustantivo o verbo, si tiene un adverbio como modificador elegirá está última opción. Creemos que merece destacar lo pragmático de la definición, en vez de buscar un concepto lingüístico Sinclair define el "producto" almacenado como "pieces of languaje", es decir pedazos o retazos de lengua escrita. Por lo tanto, un corpus puede contener palabras aisladas, frases o expresiones de uso habitual que incluso pueden no estar completas-

La Real Academia Española de la Lengua (RAE), en su 23 edición, define corpus como conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación." La palabra es de origen latino pero nos ha llegado a través del inglés, donde conserva las formas singular "corpus" y plural "corpora". La RAE (2015) junto con las correspondientes hispanoamericanas y algunas fundaciones están desarrollando el mayor corpus del español actual, llamado Corpes XXI, a fecha de 17 de noviembre de 2015 la versión 0.82 cuenta con 222.080 documentos, la mayoría textos.

La riqueza de una base de datos como se puede medir por diversos factores como son: el número de archivos que la componen, el número de frases y palabras distintas de entrenamiento, el número de palabras y caracteres totales, el promedio de caracteres por palabra, el número de horas de audio y cuántas de ellas son de silencio y voz, la cantidad de frases de hombres y mujeres, el número de hablantes distintos, el SNR promedio (signal to noise ratio, razón entre señal y ruido), el pitch promedio (que valora las frecuencias más relevantes y es el mejor criterio para distinguir hombres y mujeres).

Con frecuencia en la literatura se usa el término léxico como sinónimo de corpus. La palabra léxico tiene varias acepciones, pero la que nos interesa es la que lo define como una base de datos donde todas las palabras de una lengua o todo el vocabulario de un hablante o jerga se categoriza según sus características, y su origen puede ser tanto una fuente externa como interna al texto. Según esta concepción el corpus sería un subtipo de léxico.

2.4 Análisis del discurso

Entendemos como análisis del discurso el conjunto de metodologías y tecnologías capaces de abordar estudios sociales, culturales, antropológicos a través de los corpus textuales o lingüísticos. Estas herramientas tecnologías sirven para comprender el significado y extraer concomimiento sobre el mismo dentro de un contexto, de sentencias en el texto del dominio, de los interlocutores y sus intenciones, de las relaciones entre ellos y de las relaciones con el mundo.

El instituto cervantes define el análisis del discurso como a una disciplina cuyo objeto de estudio es el discurso, esto es, el uso que de la lengua hacen los hablantes en unas situaciones determinadas. De este modo, la totalidad de enunciados de una sociedad, bien sean orales o escritos, se convierte en objeto de estudio.

2.5 iSisad

El propósito de este TFG es el de servir como vehículo de entrada a una herramienta de análisis del discurso desarrollada por los profesores Josú Cruz, Julio Brito y Francisco Martínez.

La técnica aplicada para dicho análisis consiste en lematizar las palabras del corpus y asignarle grupos nocionales. Dichos grupos nocionales indican a qué se refiere la palabra, es decir si está nombrado a personas, animales,

lugares cosas materiales o ideas intangibles, indicando lugar o tiempo y un largo etcétera de descriptores que se le asigna a cada lema.

Se trata de un trabajo largo porque en el método actual se lematiza de forma manual y es por ello que se está desarrollando la herramienta para automatice lo más posible la asignación de grupos nocionales mediante el aprendizaje y técnicas metaheurísticas.

Es presente TFG es un avance en la entrada de los corpus puesto que la herramienta está pensada para el análisis de encuestas previamente registradas como audio.

El estado actual de la herramienta permite almacenar los corpus, descomponer en palabras, eliminar automáticamente las palabras vacías de contenido, pre-lematizar automáticamente y además proporciona herramientas para facilitar la asignación de los grupos nocionales.

A continuación se muestran unas capturas de ejemplo de funcionamiento de la herramienta iSisad:

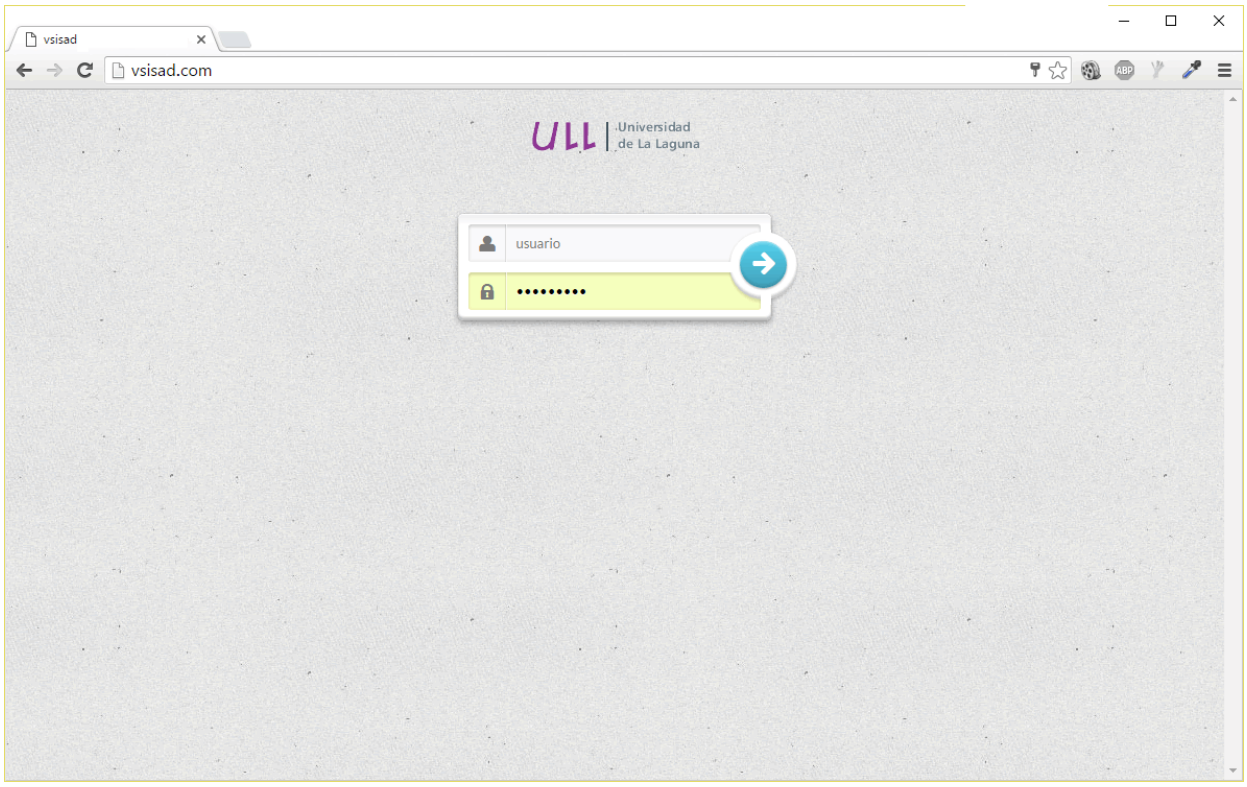


Figura 1: Login

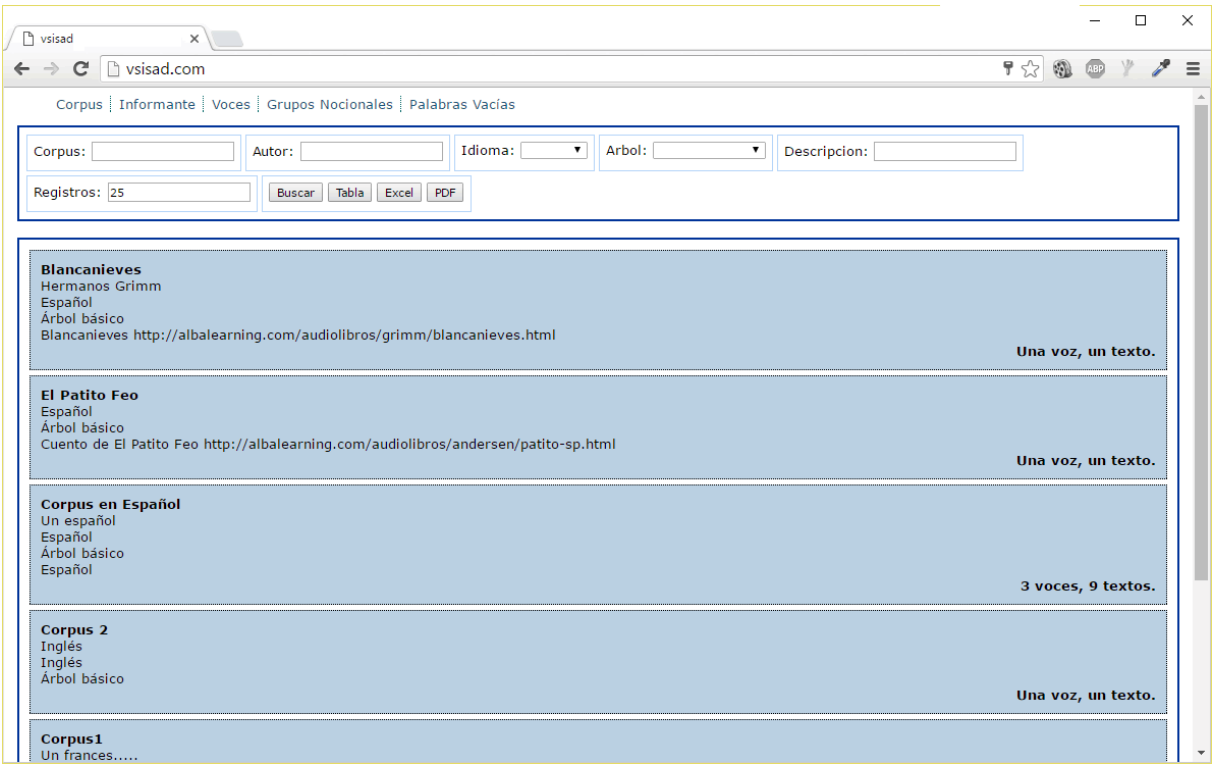


Figura 2: Selección de corpus

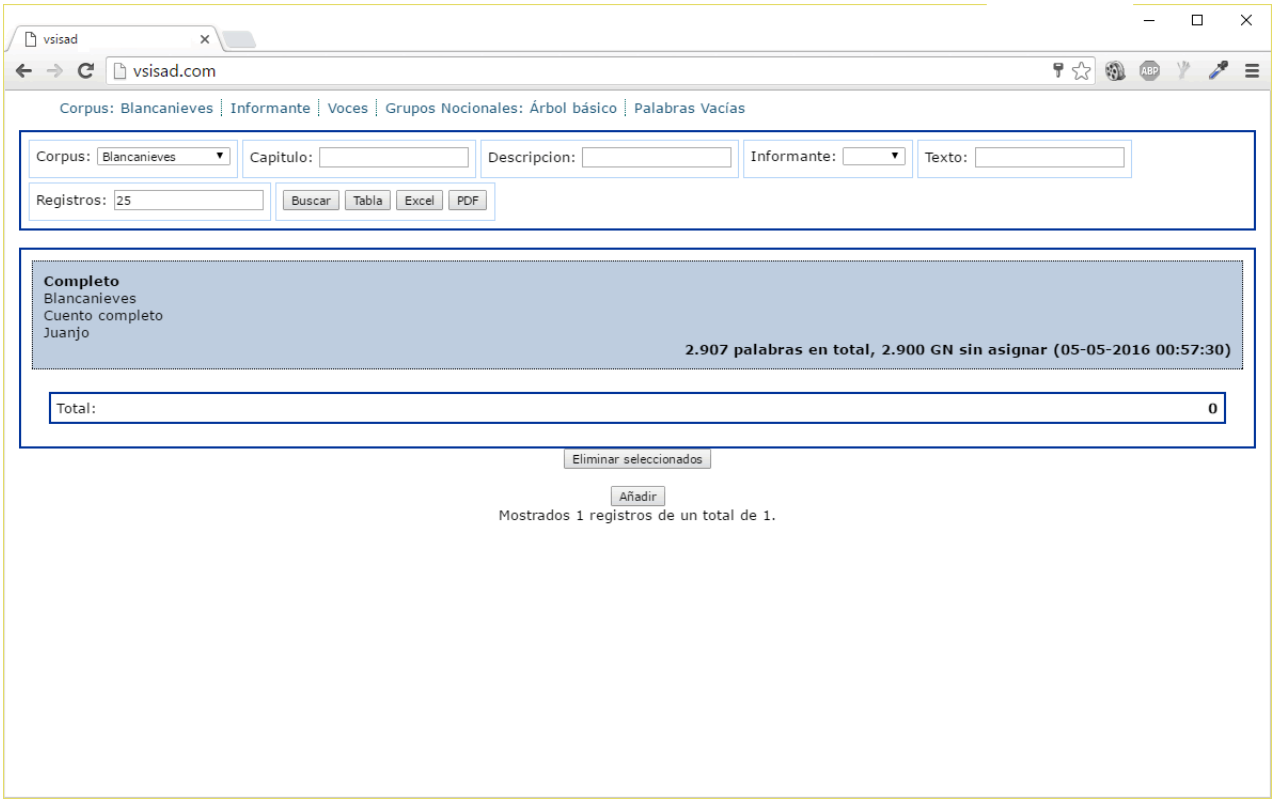


Figura 3: Selección de texto

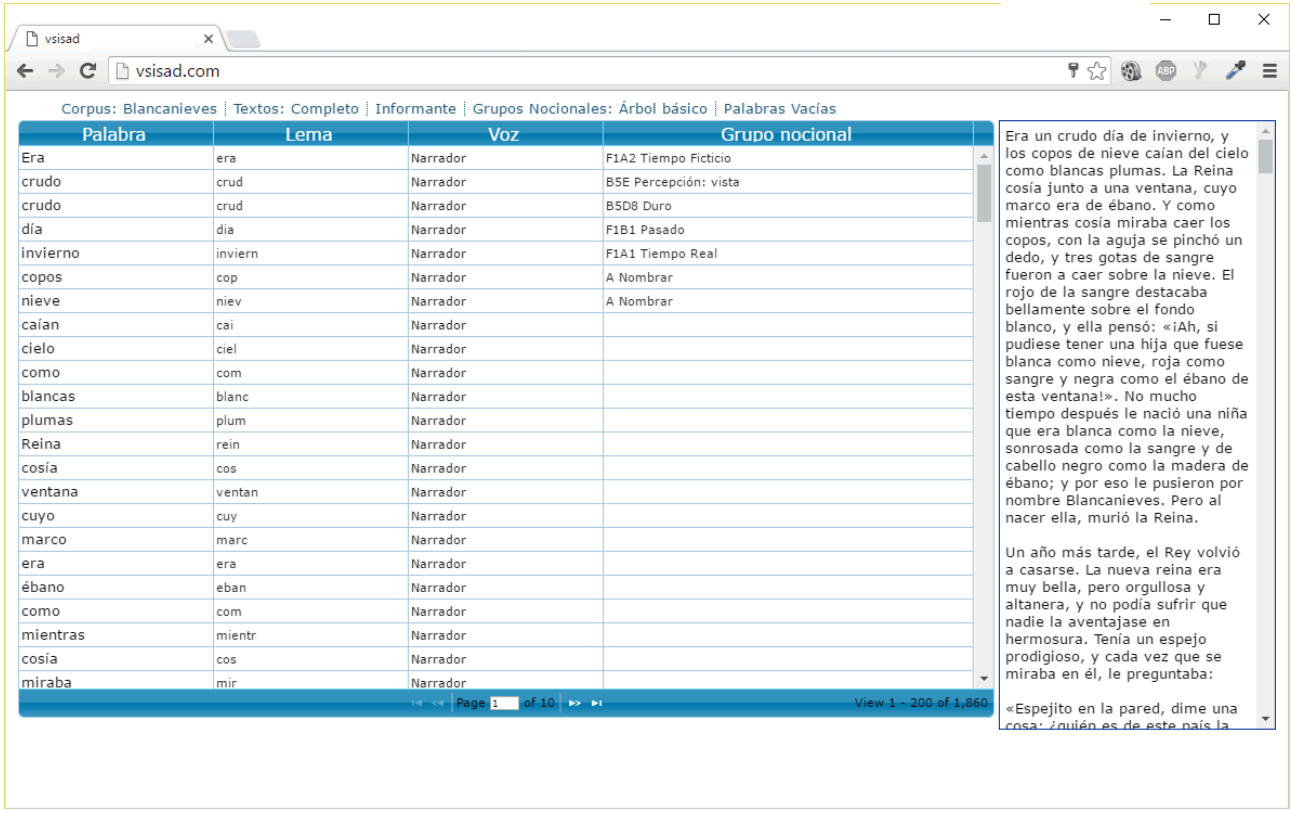


Figura 4: Herramienta de lematizado

Figura 5: Ejemplo de etiquetado en forma de nube de conceptos

Capítulo 3.
Estado del arte en SRAH o ASR

3.1 Concepto de SRAH

Es más complicado definir en qué consiste un SRAH que en tener una idea intuitiva del mismo. Mediante el reconocimiento del habla se convierten palabras en objetos que una máquina puede leer (Jurafsky y Martin, 2012) Llamas y Cardeñoso (1997 pp. 15) lo definen "desde un punto de vista práctico" como: "el proceso por el que un modelo de cómputo es capaz de traducir fielmente los sonidos asociados con una unidad de discurso y codificarlos en alguna forma de secuencia simbólica que representa total o parcialmente la carga conceptual del mensaje hablado."

Destacamos que estos autores no dicen que el producto final sea exclusivamente un texto, hablan de una secuencia simbólica, porque puede ser imprescindible que genere, como proceso previo a la identificación de un texto, una base de datos de vocabulario, sin olvidar que a veces puede ser muy útil que genere directamente una orden.

Explicar la utilidad de estos sistemas se escapa del ámbito de este trabajo, es una percepción individual. Son útiles porque facilitan que el operador interactúe con la máquina en un entorno natural, usando sólo la voz.

Resumen histórico:

Los logros de los sistemas RAH han evolucionado desde su inicio en los años 30 y 40 en que se reconocen fonemas aislados, dependientes de locutor, hasta los sistemas actuales que buscan reconocer el lenguaje natural en cualquier actividad humana y en condiciones desfavorables (Sani, P y Kaur, P. 2013; Huang y cols. 2014)). La institución que más ha patrocinado el desarrollo de estos programas es el sistema defensivo de los EE. UU,

principalmente a través de su proyecto Defense Advanced Research Program (DARPA).

El interés por los sistemas RAH se inició en la década de los 40 y mediante estudios físicos, sin participación de ordenadores. Hasta los años 50 no se logró un RAH que identificara dígitos, pero era específico del locutor. En los años 60 los avances en el campo de la lingüística facilitaron el procesamiento del habla. En los 70 se empezaron a aplicar los modelos estadísticos y se desarrollaron patrones de reconocimiento. En los 80 la tecnología ya está asentada y se reconocen palabras encadenadas, no vocablos aislados, se pasa de la mera comparación con plantillas a usar ya los modelos estadísticos, sobre todo se extendió el uso de los Modelos Ocultos de Markov. Estos modelos mejoran la capacidad de los sistemas, pues a partir de los parámetros observables permite definir la probabilidad estadística de parámetros desconocidos u ocultos; es muy útil en reconocer patrones.

A finales de los 80 y en los 90 se investiga sobre todo en reconocimiento del habla continua, para ello se mejoran los algoritmos utilizando como modelos las redes neuronales, además se han grabado grandes bases de datos, se han mejorado los sistemas de decodificación fonética y esto ha permitido trabajar en condiciones más naturales, incluyendo el ruido.

En los 90 se han mezclado las ventajas de los HMM con las redes neuronales. En la actualidad, los sistemas son cada vez más exactos y potentes, más fiables; pero la fiabilidad también se gana en función de que cada vez son más específicos como, por ejemplo, el dictado de informes radiológicos.

3.2 Tipos de SRAH actuales y futuros

La complejidad de los sistemas RAH sigue un curso paralelo a su desarrollo en el tiempo. Así, de más sencillos a más sofisticados tenemos (Ghai, W y cols. 2012):

1. **Reconocimiento de comandos o palabras aisladas**: Estos sistemas se caracterizan por usar un vocabulario pequeño, una gramática más sencilla y rígida que el lenguaje natural, gramática de

estados finitos, y un modo de hablar preestablecido en forma de palabras aisladas. Por lo tanto son sistemas que se permiten ser más independientes del locutor, tolerar mejor el ruido y de respuesta más sencilla y rápida. Los requerimientos técnicos son inferiores (menos velocidad de procesado, memoria, calidad de altavoces...). Son ideales para usarlos en aplicaciones de mando y control, como las que se usan en domótica u ortesis para minusválidos.

2. **Reconocimiento de palabras conectadas:** Aceptan palabras de una en una, separadas por pausas. Se puede usar un lenguaje planificado continuo. Su principal inconveniente es que es un sistema limitado en las posibilidades que ofrece, no está diseñando para reconocer habla en una situación de vida diaria.

3. **Reconocimiento del habla continua:** A efectos de nuestro trabajo consideraremos habla continua o lenguaje natural o espontáneo al que realizan los seres humanos en condiciones habituales de la vida diaria. Los sistemas que la reconozcan van a tener requerimientos técnicos evidentemente muy superiores a los de reconocimiento de comandos. Estos sistemas nos liberarían del teclado, que es la interface más común de interacción hombre-máquina. Hay autores que distinguen entre reconocimiento del habla continua, en que las palabras se pronuncian sin marcar las pausas y es el propio sistema el que tiene que identificar los límites, frente a reconocimiento del lenguaje espontáneo o natural, en un ambiente ruidoso, no controlado y sin ensayo o entrenamiento por parte del hablante.

Los dos primeros suelen, o tal vez es mejor decir solían, ser dependientes del hablante y por lo tanto necesitaban ser entrenados. El objetivo de diseño de los actuales es que sean independientes del hablante (sexo o edad) e incluso independientes del estado anímico o de salud del hablante.

3.3 Problemas y desafíos comunes de los SRAH

En teoría el sistema ideal debería ser capaz de trabajar en tiempo real, ser independiente del hablante, ser preciso y contener un diccionario de algunos cientos de miles de palabras (Kacur, 2008)

Si bien las posibles aplicaciones de los SRAH sólo tienen como límite nuestra imaginación en el camino nos topamos con importantes dificultades que se están investigando para soslayarlas son, entre otras:

a. necesitan entrenamiento por parte del usuario, los sistemas que no precisan entrenamiento tienen un pobre corpus acústico y por lo tanto tienen menor rendimiento.

b. son dependientes de la característica del habla, así el mismo español tiene una diferente pronunciación en Canarias, Méjico, Argentina o Madrid.

c. No reconoce los modismos o particularidades regionales.

d. Se afectan por el ruido, por ejemplo varios locutores hablando a la misma vez.

e. Necesitan un gran volumen de datos o corpus si queremos que reconozca lenguaje natural.

3.4 Arquitectura general de un SRAH

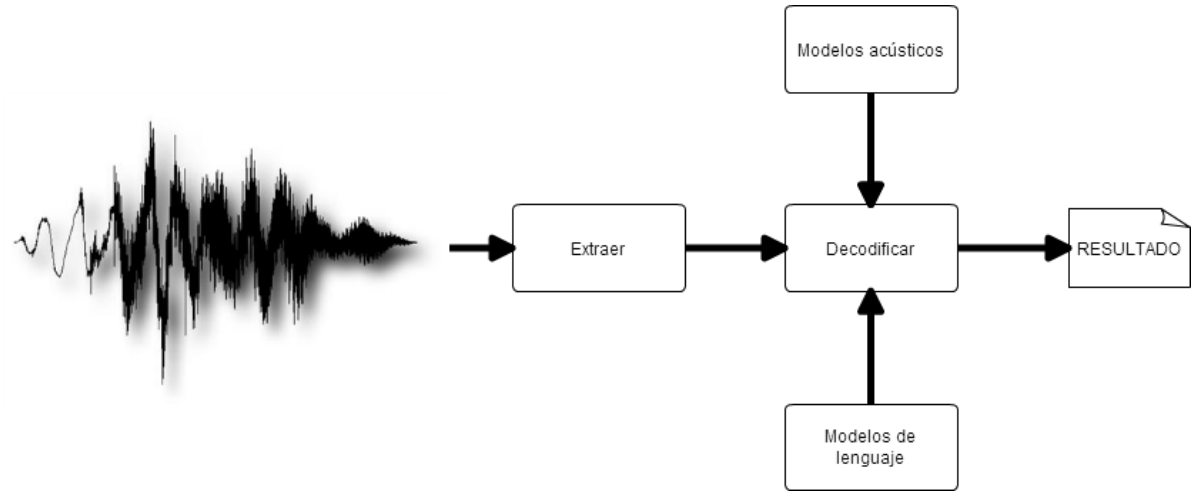


Figura 6: Esquema general de un SRAH

Los esquemas de funcionamiento de los SRAH se han ido complicando a lo largo de la historia de forma paralela como ha aumentado su capacidad de reconocimiento (Ghai, W y cols. 2012, Saini y Kaur, 2013) A efectos de

nuestra exposición trataremos un modelo ideal y al explicar los productos que hemos probado haremos referencia a sus diferencias específicas. En ocasiones tenemos varias alternativas que se suelen corresponder con fases históricas en su desarrollo, aunque un sistema más antiguo no significa que no sea útil para aplicaciones específicas, pero no cumplen los criterios para llamarlos SRAH continua o natural. En los siguientes puntos, describimos las partes fundamentales que componen dicha arquitectura.

3.4.1 Preprocesado o adquisición de la señal acústica.

Como su nombre indica es la fase en la que se adquiere la señal acústica, además se extraen sus características. El paso inicial es la transformada rápida de Fourier que permite digitalizar la señal, posteriormente existen múltiples tecnologías que delimitan las características del ruido, extraen las características adaptativas y discriminatorias y que corrigen las diferencias entre locutores o las diferencias que un mismo locutor presenta en su discurso en función de su estado anímico, de salud o edad (Therese, S. y Lingam, C. 2013).

3.4.2 Reconocimiento de patrones acústicos.

Los patrones se pueden generar siguiendo un modelo acústico-fonético o usando tecnologías de reconocimiento de patrones, con ellos se compararan nuestra señal ya pre-procesada. Como es lógico pensar los primeros trabajos se hicieron aproximaciones acústico-fonéticas; en esencia se basan considerar que el número de unidades fonéticas es limitado, si las caracterizamos a todas y las identificamos podremos comparar con nuestra voz o sonido problema. La realidad resultó ser que su número puede ser astronómico y difícil de localizar, de hecho no ha generado aplicaciones comerciales viables. Los SRAH que usan esta tecnología siguen la secuencia: análisis espectral, detección de las características, segmentación e identificación de las unidades fonéticas y al final reconocimiento al comparar con la base de datos de plantillas. Los modelos basados en reconocimiento de patrones son más robustos y permiten una calidad superior de reconocimiento. Se basan en crear un vector patrón cuyas características mejor se adapten con el vector de entrada. Esta fase a su vez se divide en entrenamiento y comparación. En la fase de entrenamiento se genera un patrón que se puede modificar en el tiempo. La

comparación se hace entre nuestra señal a estudiar y el modelo patrón. El sistema escogerá el patrón que tiene más probabilidades de ser el adecuado, el patrón puede ser cualquier sonido, una palabra aislada o una frase. Hay dos aproximaciones para crear la base de datos cuando reconocemos patrones. Uno es usando un modelo de plantillas y otro usando un modelo estocástico. Cuando se usan plantillas se elige un prototipo de habla y se almacena como patrón de referencia, al que también llamamos corpus. De esta manera se disminuyen los errores de segmentación y clasificación de los sonidos, pero exige que cada palabra tenga su propia plantilla de referencia. Tendremos un sistema potente, que necesita una gran cantidad de capacidad de computación y que es muy caro y laborioso para configurar. El modelo estocástico usa modelos probabilísticos, permite distinguir ruidos, palabras homófonas y las variables propias del hablante, la técnica más potente y eficaz la llamada Modelos Ocultos de Markov (HMM, siglas en inglés de HiddenMarkovModels). En el que la unidad a identificar tiene forma de vector que se compara con un vector de referencia, con el que tenga una distancia mínima y por lo tanto sea el más parecido. Posteriormente se ha mejorado añadiéndole algoritmos que tengan en cuenta las densidades gaussianas, de manera que las frecuencias que contribuyen a definir el tono estén más representadas, por ejemplo, que las que definen el timbre; el estándar fueron los coeficientes cepstrales de frecuencias mel. Esta tecnología es muy eficaz y se sigue usando en la actualidad, su ventaja es que introduce una variante piscofisiológica, pues las unidades mel se definieron a partir de la capacidad discriminativa del oído humano (Stevens y Volkmann, 1940). Existen otras técnicas y otros algoritmos que intentan hacer y mejorar esa función. Más recientes es el uso de redes neuronales profundas, que imitan al cerebro y funciona por capas, muy útil para descubrir patrones ocultos y que son capaces de sustituir al modelo HMM-con densidades gaussianas. Han surgido sistemas híbridos que combinan redes neuronales y HMM que disminuyen significativamente el error de decodificación.

3.4.3 Reconocimiento de modelos de léxico.

El SRH también debe contar con un diccionario o lexicom, en forma de corpus, modelado y categorizado. El diccionario se puede optimizar en función del tamaño y del interés del diseñador. Estos diccionarios han sido

esenciales pero los modelos actuales necesitan gran volumen de datos para aprender y modelarse lo que hace necesario enormes cantidades de datos que ya no pueden estar tan estructurados. Contienen la pronunciación de cada palabra.

3.4.4 Reconocimiento de modelos de lenguaje.

El SRH ideal debe tener unas reglas sintácticas y semánticas complejas. De esta manera elegirá o aceptara la transcripción correcta con más probabilidad. Los modelos más usados son la gramática cerrada de estados finitos, que restringe las posibilidades, y el modelo de N-gramas, que busca los candidatos más probables. La mayoría de los sistemas actuales utilizan modelos de N-gramas que se están mejorando con modelos de redes neuronales profundas

3.4.5 Decodificación.

También llamado reconocimiento o decisión. El sistema acepará una posible versión con respecto a otra como la más probable, basándose tanto en el modelo acústico como en el modelo de lenguaje. En la actualidad se usan sobre todo modelos probabilísticos en los que, como hemos dicho, el ordenador selecciona el vector de su base de datos que más se asemeja al vector entrada, nos puede dar incluso varias alternativas. Por lo tanto el texto generado tiene una determinada probabilidad de representar el mensaje original.

3.5 Rendimiento de un SRAH

Para comparar SRAH se suelen medir su exactitud y velocidad (Chen, 2016). La exactitud se mide sobre todo con el test de error de palabras (WER del inglés Word Error Rate) y la velocidad por el índice de tiempo real. El WER compara el texto generado con el texto origen y cuenta el número de errores (sustituciones, borrados e inserciones) y se divide por el número total de palabras. El índice de tiempo real relaciona el tiempo de procesado con el tiempo de introducir el dato. Si la relación es menor o igual a 1 entonces estamos hablando de procesamiento en tiempo real.

Para realizar un test WER en un sistema de reconocimiento del habla natural podemos tardar horas o días, un sustituto útil, por su rapidez, en la fase de desarrollo sería un control interno, no tan informativo y completo como el externo WER. Con este propósito se ha desarrollado el valor de perplejidad (perplexity), que mide el número medio de palabras que siguen a una determinada (Jurafsky, 2016).

Capítulo 4.
Análisis y evaluación de sistemas de reconocimiento del habla.

En este capítulo analizaremos y evaluaremos los sistemas de reconocimiento del habla sobre todo la técnica DAFO. Hemos usado los programas de licencia libre que nos permiten RAH y hemos diseñando script o programas de enlace entre nuestro problema y el programa objeto de prueba. El trabajo de programación que hemos llevado a cabo es el de adaptar los diferentes sistemas más punteros en este campo, como es el caso de Microsoft y Google en sistemas cerrado, o CMUSphinx y Julios en sistemas open source. En todos los casos hemos usado como plantilla el código ejemplo que proporcionan, llevando a cabo su configuración correspondiente y sobre todo la modificación de ese código para que se adapte a nuestras necesidades. Con el único fin de facilitar la lectura de este trabajo hemos decidido agrupar dichos programas como apéndices al final del mismo.

4.1 Técnica DAFO

La técnica DAFO se desarrolló como una herramienta de gestión empresarial, para orientar la toma de decisiones; pero ha demostrado ser útil para estudiar cualquier tipo de organización como por ejemplo el sistemas sanitario o el educativo (CommunityTool Box, 2016). Es más, también se ha demostrado su utilidad para tomar decisiones de tipo personal. DAFO es el acrónimo de sus cuatro componentes (Debilidades, Amenazas, Fortalezas y Oportunidades), en ocasiones también se llama análisis o matriz FODA. El nombre inglés es SWOT (Strength, Weakness, Opportunity, Threat) y tiene la ventaja de hacer dos parejas de características, por un lado las internas

(Fortalezas y Debilidades) y por otro las externas (oportunidades y amenazas). Otro de sus usos, muy pertinente a nuestro trabajo, es organizar la información que proviene de fuentes diferentes. En nuestro caso que trabajamos con información podemos ver que aparte de la pareja interno-externo tenemos otra de connotaciones positivas, fortalezas y oportunidades, junto a otra de connotaciones negativas, debilidades y amenazas (Management SciencesforHealth y UNICEF, 2016) Por supuesto no podemos olvidar que las técnicas DAFO son subjetivas (Team FME, 2013), pero los objetivos y el método de trabajo de nuestro estudio llevan implícitos cierto carácter de subjetividad que intentaremos compensar con un trabajo estructurado además de buscar fuentes bibliográficas diversas y actuales.

4.2 API de Google

Google nos permite la posibilidad de usar su API de reconocimiento de voz, diseñada como parte de una estrategia comercial que pretende contrarrestar el poder de la de Apple y con el propósito de ser usada en sus dispositivos portátiles ("wearables" en el nomenclátor actual).

Google nos facilita dos formas de usar su SRAH. Una, la "Web Speech Api Demostration", está disponible en 40 idiomas y se puede acceder en línea a través de la dirección https://www.google.com/intl/es/chrome/demos/speech.html. Su aspecto es muy sobrio. Google ofrece un código que incrusta un cuadro de texto sobre el que se volcará el audio, en forma de texto, que ha conseguido transcribir y algunos controles para manejar el inicio / detención de la captura del audio. Es un sistema ideal para dictado, pero en nuestro caso deseamos disponer del texto desde un audio "enlatado" (pregrabado en formato MP3) para ello es posible puentear la entrada del micrófono desde la salida de los altavoces (con un cable Jack macho en ambos terminales) o por medio de software de virtualización. Tiene la ventaja de no tener limitaciones en el tiempo de grabado pero la desventaja de que no es modificable por el desarrollador, por lo que no entra dentro de los presupuestos que habíamos establecidos para ser valorada.

La otra API de Google es la “Google Speech API V2”. Que permite más autonomía al desarrollador porque se puede configurar para adaptarse a sus necesidades, pero es una versión totalmente en prueba, únicamente para desarrolladores identificados y uso personal. Para conectarnos con “Google Speech API V2” tenemos varias opciones, nosotros vamos a realizarla por medio de un “script Shell” que nos va a permitir convertir un fichero de audio en un fichero de texto.

EL único requisito que nos pide Google es una clave de desarrollador (“Api Key") que podemos obtener tras darnos de alta en el sitio desarrolladores de proyectos Chromium.

Limitaciones, como hemos dicho la API está en su versión de prueba, por lo que no admite ficheros de más de 15 segundos ni más de 50 llamadas a la API por día.

El “scrip Shell” que hemos programado lo hemos colocado en el apartado de apéndices.

4.2.1 Experiencia de uso

	A	B	C
Eficacia(*)	95%	85%	75%

Tabla 1: Experiencia de uso Google

* Medimos la eficacia como el tanto por ciento del texto que ha reconocido sin cometer errores respecto al total. Ver las diferentes entradas en el Apéndice B.

4.2.2 Análisis DAFO

Fortalezas	Debilidades
- Gratuita - Alta eficacia en óptimas condiciones.	- Limitación de 15 segundos por llamada y 50 llamadas a la API diarias. - Sin subscripción de pago para ampliar las debilidades. - No identifica entre hablantes.
Oportunidades	**Amenazas**
- Aunque el producto sea un servicio Web y por lo tanto estemos sujetos a disponibilidad, tiene un crecimiento exponencial en eficacia puesto que se nutre de los millones de usuarios de la red.	- Google no garantiza el funcionamiento. Al ser un servicio Web y si el producto queda descontinuado, nuestro sistema quedará inservible.

Tabla 2: Análisis DAFO Google

4.3 API de Microsoft

Microsoft es una API, como la de Google, basada en la nube que nos proporciona algoritmos para procesar el lenguaje hablado. Está API está dentro de un conjunto de tecnologías que Microsoft ofreció a los desarrolladores, denominado Proyecto Oxford. La idea principal de este proyecto, es permitir a los desarrolladores crear aplicaciones más inteligentes, sin necesidad de preocuparse por el aspecto de aprendizaje de la máquina. La serie de servicios está actualmente disponible para uso gratuito, previo registro en la plataforma Microsoft Azure, limitado en fase beta.

A diferencia de la API de Google, Microsoft tiene dos modos de reconocimiento de voz:

- Frase corta: esta consta de 15 segundos de duración. Los datos se envían al servidor y el cliente va a recibir múltiples resultados parciales y una N-mejor resultado final de múltiples alternativas.
- Frase larga: el enunciado puede ser de 2 minutos de duración. A medida que los datos se envían al servidor, el cliente recibe múltiples resultados

parciales y múltiples resultados finales. Estos resultados tienen detección de párrafos y signos de puntuación.

La limitación principal es el tiempo continuado de grabación, y la limitación de las llamadas a su servicio.

En el Apéndice A.2, está el código de la aplicación que ha sido desarrollado con .NET, pero se podría programar en otros lenguajes como JAVA.

4.3.1 Experiencia de uso

	A	B	C
Eficacia(*)	95%	85%	75%

Tabla 3: Experiencia de uso Microsoft

* Medimos la eficacia como el tanto por ciento del texto que ha reconocido sin cometer errores respecto al total. Ver las diferentes entradas en el Apéndice B.

4.3.2 Análisis DAFO

Fortalezas	Debilidades
- Gratuita - Alta eficacia en óptimas condiciones - Detección de párrafos	- Limitación de 2 minutos por llamada. - 20 transacciones por minuto y 5000 mensuales. - Sin subscripción de pago para ampliar las debilidades. - No identifica entre hablantes.
Oportunidades	**Amenazas**
- Servicio de computación en la nube, alimentado por los millones de personas que usan Windows diariamente.	- La API está en desarrollo, existen funciones cambiantes.

Tabla 4: Análisis DAFO Microsoft

4.4 Otros sistemas open source

Hay varios sistemas open source, que permiten desarrollar en su totalidad, un SRAH. Los más importantes son CMUSphinx y Julius.

- Julius es un sistema de software libre con licencia BSD, desarrollado en Japón. Su última versión es de 2011.
- CMUSphinx es un conjunto de herramientas desarrolladas por la Universidad de CarnnegieMellon. Permite su implantación en diferentes lenguajes, incluso tiene una versión que permite su uso en sistemas embebidos. Y a diferencia de Julius su última versión es mucho más actual, de diciembre de 2015.

Estos sistemas actualmente son útiles para reconocer unas pocas palabras, por ejemplo, un software de control de una casa, en el que se le dan pocas instrucciones, como: "encender luz", "cerrar ventana",... No son útiles para dictados, ya que en un dictado no se especifica una gramática limitada, sino que el usuario podría decir casi cualquier cosa, para esto se usan un corpus.

El gran problema que tiene el reconocimiento de voz es que no hay forma libre de tener un corpus decente. El proyecto más conocido que nos permite tener un corpus libre es VoxForge, el cual tiene como objetivo recoger la voz que la gente done, para compilar un modelo acústico que usarán las herramientas que hemos mencionado en este apartado.

Capítulo 5.
Conclusiones y líneas futuras

La API de Google y la API de Microsoft, en nuestro modelo experimental se han mostrado equivalentes en cuanto a identificar un hablante por voz enlatada, grabada y con ruido ambiente. La CMUSphinx a pesar de que en la teoría puede parecer que tiene ventajas superiores, en la práctica, y en nuestras condiciones experimentales se ha mostrado muy inferior.

Las API de Google y de Microsoft limitan el tiempo de uso gratuito a 15 segundos, que puede ser suficiente para que un desarrollador lance una instrucción pero insuficiente para trabajar grandes cantidades de texto. Microsoft solventa, sólo en parte, esta limitación porque permite reconocimientos de hasta 2 minutos, con identificación de párrafos y signos de puntuación, CMUSphinx no tiene limitación de tiempo.

La API de Google y Microsoft están diseñadas para ser usadas por desarrolladores, su software no es open source y por lo tanto no podemos ni controlar ni tener conocimientos básicos de cómo se controla el sistema. La CMUSphinx es más versátil, permite adaptarse a necesidades específicas porque usa un software modular, de libre acceso.

Con respecto a qué nos puede deparar el futuro volvemos a la introducción a este trabajo. El límite es nuestra imaginación. El futuro será una interface hombre máquina, independiente del hablante, del idioma y del ruido ambiente, que traduzca en tiempo real; grupos de personas de diferentes edades y culturas compartiendo conocimientos sin verse limitados por el idioma. Para llegar a esa fase necesitaremos:

- Definir el mejor modelo lingüístico, y aquí incluimos corpus, léxico, gramática, adaptado a las jergas y variantes locales del habla.

- Miniaturización del hardware que lo haga fácil de trasportar.
- Aumento de la velocidad de proceso, los ordenadores deben ser capaces de procesar millones de datos por segundo, sin aumentar su tamaño ni su consumo energético
- Aumento de la velocidad de transferencia de datos. Con la tecnología actual y la prevista en las próximas décadas es imposible que un sistema tan potente forme parte de un computador aislado, siempre será parte de una gigantesca red de información desde el comienzo de esta investigación, las dos grandes plataformas analizadas, Google y Microsoft, han evolucionado enormemente.

Capítulo 6.
Summary and Conclusions

In our experimental model, Google API and Microsoft API, wereequivalents in terms of identify a speaker by recorded voice, recorded and with ambient noise. The CMUSphinx, in spite of in theory it may seem that it has superior advantages, in practice, and in our experimental conditions it has been far lower.

Google and Microsoft APIs limit the free airtime to 15 seconds, which may be sufficient for a developer to launch an instruction but insufficient to work with large amounts of text. Microsoft solved only in part, this limitation because it allows recognitions up to 2 minutes, identifying paragraphs and punctuation marks, while CMUSphinx has no time limit.

Google and Microsoft APIs are designed to be used by developers. Its software is not an open source and therefore we can neither control nor have basic knowledge of how the system is controlled. The CMU Sphinx is more versatile, it allows to be adapted to specific needs because it uses a modular software, open access.

Regarding to what the future may hold for us we turn back to the introduction of this work. The limit is our imagination. The future will be a man – machine interface, independent of speech, language and ambient noise, which translates in real time; groups of people of different ages and cultures sharing knowledge without being constrained by the language. To reachthisstagewewillneed:

- Define the best language model, and here we include corpus, lexicon, grammar, adapted to the jargon and local variants of speech.
- Miniaturization of the hardware to do it easy to transport.
- Increase the speed of processing. Computers must be able to process millions of data per second without increasing its size or power consumption.
- Increase data transfer speed. With current technology and the one projected for the coming decades it is impossible for such a powerful system to be part of an isolated computer, it will always be part of a huge network of information. Since the beginning of this research, the two major platforms analyzed, Google and Microsoft, have evolved tremendously.

Capítulo 7. Presupuesto

Este capítulo es obligatorio. Toda memoria de Trabajo de Fin de Grado debe incluir un presupuesto.

7.1 Planificación:

Recursos humanos: todo el análisis y la programación han sido desarrollados por el alumno bajo la supervisión del director del trabajo, por lo tanto no se han generado costes.

7.2 Recursos materiales:

- Ordenador de sobremesa: Apple MacBook Pro Retina 13 pulgadas, 1440 euros, suponiendo, según legislación vigente, una amortización mínima de 4 años y una dedicación del 50% durante el último años: 180 euros

- Programas de procesamiento de texto:

Programa	Precio	Descargado
Sublime Text3	0€	https://www.sublimetext.com/3
Eclipse Mars	0€	https://eclipse.org/
Visual Studio Community 2015	0€	https://www.visualstudio.com/

Tabla 5: Programas de procesamiento de texto

- Programas motivo de estudio:

Programa	Precio	Descargado
CMUSphinx	0€	http://cmusphinx.sourceforge.net/
Speech Api	0€	https://www.projectoxford.ai/
Google Api V2	0€	http://www.chromium.org/developers
Audacity 2.1.2	0€	http://www.audacityteam.org/
PHP	0€	https://secure.php.net/
MySQL	0€	https://www.mysql.com/
Apache	0€	https://httpd.apache.org/
Servidor Web ULL	0€	http://www.ull.es/
iSisad	No definido	-

Tabla 6: Programas motivo de estudio

- Resumen recursos económicos:

Recursos humanos	0€
Ordenador	0€
Programas	0€
Total	180€

Tabla 7: Tabla resumen de los Tipos

Apéndice A.
Códigos de los algoritmos

A.1. Algoritmo Api Google

```
#!/bin/bash
# Usage info
show_help(){
cat << EOF
  Usage: ${0##*/} [-h] [-i INFILE] [-d DURATION] [-r RATE] [-l LANGUAGE] [-k KEY]
  Record an utterance and send audio data to Google for speech recognition.

       -h|--help               display this help and exit
       -i|--input     INFILE   use INFILE instead of recording a stream with parecord.
       -d|--duration  FLOAT    recoding duration in seconds (Default: 3).
       -l|--language  STRING   set transcription language (Default: en_US).
                               Other languages: fr-FR, de-DE, es-ES, ...
       -r|--rate      INTEGER  Sampling rate of recorded data (Default: 16000).
                               If -i|--input is used, the sampling rate must be
supplied by the user.
       -k|--key       STRING   Google Speech Recognition Key.

EOF
}

DURATION=15
LANGUAGE=es-ES
KEY=AIzaSyBHac10-RtDbCfs-edsR1cPxTrvSNJ6Lzg
RATE=1600

record(){
DURATION=$1
SRATE=$2
INFILE=$3

if hash rec 2>/dev/null; then
# try to record audio with sox
rec -q -c 1 -r $SRATE$INFILEtrim 0 $DURATION
else
```

```
# fallback to parecord
        timeout $DURATIONparecord$INFILE --file-format=flac --rate=$SRATE --channels=1
fi
}

# parse parameters
while[[$# -ge 1 ]]
do
key="$1"
case$key in
       -h|--help)
show_help
exit 0
       ;;
       -i|--input)
INFILE="$2"
shift
       ;;
       -d|--duration)
DURATION="$2"
shift
       ;;
       -r|--rate)
SRATE=$2
shift
       ;;
       -l|--language)
LANGUAGE="$2"
shift
       ;;
       -k|--key)
KEY="$2"
shift
       ;;
       *)
echo"Unknown parameter '$key'. Type $0 -h for more information."
exit 1
       ;;
esac
shift
done

if[[ ! "$DURATION"]]
then
echo"ERROR: empty or invalid value for duration."
exit 1
fi
```

```
if[[ ! "$LANGUAGE"]]
then
echo"ERROR: empty value for language."
exit 1
fi

if[[ ! "$INFILE"]]
then
INFILE=record.flac
if[[ ! "$SRATE"]]
then
=16000
fi
echo"Say something..."
echo""
      record $DURATION$SRATE$INFILE
else
if[[ ! "$SRATE"]]
then
>&2 echo"ERROR: no sampling rate specified for input file."
exit 1
fi

echo"Try to recognize speech from file $INFILE"
echo""
fi

RESULT=`wget -q --post-file $INFILE --header="Content-Type: audio/x-flac;
rate=$SRATE"-O - "http://www.google.com/speech-
api/v2/recognize?client=chromium&lang=$LANGUAGE&key=$KEY"`

FILTERED=`echo"$RESULT" | grep "transcript.*}" |
sed's/,/\n/g;s/[{,},"]//g;s/\[//g;s/\]//g;s/:/: /g' | grep -o -i -e "transcript.*" -e
"confidence:.*"`

if[[ ! "$FILTERED"]]
then
>&2 echo"Google was unable to recognize any speech in audio data"
else
echo"Recognitionresult:"
echo""
echo"$FILTERED"
fi

exit0
```

A.2. Algoritmo API Microsoft

```
usingSystem;
usingSystem.Configuration;
usingSystem.Diagnostics;
usingSystem.IO;
usingSystem.Windows;

usingMicrosoft.ProjectOxford.SpeechRecognition;
usingSystem.IO.IsolatedStorage;
usingSystem.Runtime.CompilerServices;
usingSystem.ComponentModel;
usingSystem.Threading;

namespaceMicrosoftProjectOxfordExample
{
/// <summary>
/// Interaction logic for MainWindow.xaml
/// </summary>
publicpartialclassMainWindow : Window, INotifyPropertyChanged
    {


string _subscriptionKey;

string _luisAppID = ConfigurationManager.AppSettings["luisAppID"];
string _luisSubscriptionID = ConfigurationManager.AppSettings["luisSubscriptionID"];

string _recoLanguage = "es-ES";

privateconststringShortWaveFile = @"audioCorto.wav";
privateconststringLongWaveFile = @"audioLargo.wav";

privateDataRecognitionClient _dataClient;

publicboolIsDataClientShortPhrase { get; set; }
publicboolIsDataClientWithIntent { get; set;  }
publicboolIsDataClientDictation { get; set; }

/// <summary>
/// The MAIS reco response event
/// </summary>
privateAutoResetEvent _FinalResponseEvent;
```

```
/// <summary>
/// Gets or sets subscription key
/// </summary>
publicstringSubscriptionKey
        {
get
            {
return _subscriptionKey;
            }

set
            {
                _subscriptionKey = value;
OnPropertyChanged<string>();
            }
        }

privatereadonlystringIsolatedStorageSubscriptionKeyFileName = "Subscription.txt";
privatereadonlystringDefaultSubscriptionKeyPromptMessage  =  "Paste  your  subscription
key here to start";

#region Events

/// <summary>
/// Implement INotifyPropertyChanged interface
/// </summary>
publiceventPropertyChangedEventHandlerPropertyChanged;

#endregion Events

/// <summary>
/// Initializes a new instance of the <see cref="MainWindow"/> class.
/// </summary>
publicMainWindow()
        {
InitializeComponent();
Intialize();
            _FinalResponseEvent = newAutoResetEvent(false);
        }

/// <summary>
//  Raises the System.Windows.Window.Closed event.
/// </summary>
/// <param name="e">An System.EventArgs that contains the event data.</param>
protectedoverridevoidOnClosed(EventArgs e)
{
if (null != _dataClient)
```

```
{
                _dataClient.Dispose();
            }
            _FinalResponseEvent.Dispose();

base.OnClosed(e);
        }

privatevoidIntialize()
        {
IsDataClientShortPhrase = false;
IsDataClientWithIntent = false;
IsDataClientDictation = false;

// Set the default choice for the group of checkbox.
            _dataLongRadioButton.IsChecked = true;

SubscriptionKey = GetSubscriptionKeyFromIsolatedStorage();
        }

/// <summary>
/// Handles the Click event of the _startButton control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="RoutedEventArgs"/> instance containing the event
data.</param>
privatevoidStartButton_Click(object sender, RoutedEventArgs e)
        {
//_logText.Text = String.Empty;
            _startButton.IsEnabled = false;
            _radioGroup.IsEnabled = false;

if (IsDataClientShortPhrase)
            {
LogRecognitionStart("short            wav            file",            _recoLanguage,
SpeechRecognitionMode.ShortPhrase);

if (_dataClient == null)
                {
                    _dataClient                                                                =
CreateDataRecoClient(SpeechRecognitionMode.ShortPhrase, _recoLanguage);
                }
SendAudioHelper(_dataClient, ShortWaveFile);
            }
elseif (IsDataClientDictation)
            {
LogRecognitionStart("long            wav            file",            _recoLanguage,
```

```
SpeechRecognitionMode.LongDictation);

if (_dataClient == null)
                {
                    _dataClient                                                                    =
CreateDataRecoClient(SpeechRecognitionMode.LongDictation, _recoLanguage);
                }
SendAudioHelper(_dataClient, LongWaveFile);
            }
elseif (IsDataClientWithIntent)
            {
if (_dataClient == null)
                {
                    _dataClient     =     CreateDataRecoClientWithIntent(_recoLanguage,
ShortWaveFile);
                }
SendAudioHelper(_dataClient, ShortWaveFile);
            }
        }

privatevoidLogRecognitionStart(stringrecoSource,                    stringrecoLanguage,
SpeechRecognitionModerecoMode)
        {
WriteLine("\n--- Start speech recognition using " + recoSource + " with " + recoMode +
" mode in " + recoLanguage + " language ----\n\n");
        }


privatevoidHelpButton_Click(object sender, RoutedEventArgs e)
        {
            Process.Start("https://www.projectoxford.ai/doc/general/subscription-key-
mgmt");
        }

/// <summary>
///     Speech recognition with data (for example from a file or audio source).
///       The data is broken up into buffers and each buffer is sent to the Speech
Recognition Service.
///     No modification is done to the buffers, so the user can apply their
///     own Silence Detection if desired.
/// </summary>
DataRecognitionClientCreateDataRecoClient(SpeechRecognitionModerecoMode,
stringrecoLanguage)
        {
DataRecognitionClientdataClient = SpeechRecognitionServiceFactory.CreateDataClient(
recoMode,
recoLanguage,
```

```
SubscriptionKey);

// Event handlers for speech recognition results
if (recoMode == SpeechRecognitionMode.ShortPhrase)
            {
dataClient.OnResponseReceived += OnDataShortPhraseResponseReceivedHandler;
            }
else
            {
dataClient.OnResponseReceived += OnDataDictationResponseReceivedHandler;
            }
dataClient.OnPartialResponseReceived += OnPartialResponseReceivedHandler;
dataClient.OnConversationError += OnConversationErrorHandler;

returndataClient;
        }

DataRecognitionClientWithIntentCreateDataRecoClientWithIntent(stringrecoLanguage,
stringwavFileName)
        {
DataRecognitionClientWithIntentintentDataClient =

SpeechRecognitionServiceFactory.CreateDataClientWithIntent(recoLanguage,
SubscriptionKey,
                                                                    _luisAppID,

_luisSubscriptionID);
// Event handlers for speech recognition results
intentDataClient.OnResponseReceived += OnDataShortPhraseResponseReceivedHandler;
intentDataClient.OnPartialResponseReceived += OnPartialResponseReceivedHandler;
intentDataClient.OnConversationError += OnConversationErrorHandler;

// Event handler for intent result
intentDataClient.OnIntent += OnIntentHandler;

returnintentDataClient;
        }

privatevoidSendAudioHelper(DataRecognitionClientdataClient, stringwavFileName)
        {
using     (FileStreamfileStream     =     newFileStream(wavFileName,     FileMode.Open,
FileAccess.Read))
            {

intbytesRead = 0;
byte[] buffer = newbyte[1024];
```

```
try
                {
do
                    {
// Get more Audio data to send into byte buffer.
bytesRead = fileStream.Read(buffer, 0, buffer.Length);

// Send of audio data to service.
dataClient.SendAudio(buffer, bytesRead);
                    } while (bytesRead>0);
                }
finally
                {
// We are done sending audio.  Final recognition results will arrive in
OnResponseReceived event call.
dataClient.EndAudio();
                }
            }
        }


/// <summary>
///     Called when a final response is received;
/// </summary>
voidOnDataShortPhraseResponseReceivedHandler(object sender, SpeechResponseEventArgs e)
        {
Dispatcher.Invoke((Action)(() =>
            {
WriteLine("--- OnDataShortPhraseResponseReceivedHandler ---");
// we got the final result, so it we can end the mic reco.  No need to do this
// for dataReco, since we already called endAudio() on it as soon as we were done
// sending all the data.

                _FinalResponseEvent.Set();

WriteResponseResult(e);

                _startButton.IsEnabled = true;
                _radioGroup.IsEnabled = true;
            }));
        }

privatevoidWriteResponseResult(SpeechResponseEventArgs e)
        {
if (e.PhraseResponse.Results.Length == 0)
            {
WriteLine("No phrase resonse is available.");
```

```
            }
else
            {
WriteLine("********* Final n-BEST Results *********");
for (inti = 0; i<e.PhraseResponse.Results.Length; i++)
                {
WriteLine("[{0}] Confidence={1}, Text=\"{2}\"",
i, e.PhraseResponse.Results[i].Confidence,
e.PhraseResponse.Results[i].DisplayText);
                }
WriteLine();
            }
        }

/// <summary>
///     Called when a final response is received;
/// </summary>
voidOnDataDictationResponseReceivedHandler(object sender, SpeechResponseEventArgs e)
        {
WriteLine("--- OnDataDictationResponseReceivedHandler ---");
if (e.PhraseResponse.RecognitionStatus == RecognitionStatus.EndOfDictation ||
e.PhraseResponse.RecognitionStatus == RecognitionStatus.DictationEndSilenceTimeout)
            {
Dispatcher.Invoke((Action)(() =>
                {
                    _FinalResponseEvent.Set();

                    _startButton.IsEnabled = true;
                    _radioGroup.IsEnabled = true;

// we got the final result, so it we can end the mic reco.  No need to do this
// for dataReco, since we already called endAudio() on it as soon as we were done
// sending all the data.
                }));
            }
WriteResponseResult(e);
        }


/// <summary>
///     Called when a final response is received and its intent is parsed
/// </summary>
voidOnIntentHandler(object sender, SpeechIntentEventArgs e)
        {
WriteLine("--- Intent received by OnIntentHandler() ---");
WriteLine("{0}", e.Payload);
WriteLine();
```

```
        }

/// <summary>
///     Called when a partial response is received.
/// </summary>
voidOnPartialResponseReceivedHandler(object sender, PartialSpeechResponseEventArgs e)
        {
WriteLine("--- Partial result received by OnPartialResponseReceivedHandler() ---");
WriteLine("{0}", e.PartialResult);
WriteLine();
        }

/// <summary>
///     Called when an error is received.
/// </summary>
voidOnConversationErrorHandler(object sender, SpeechErrorEventArgs e)
        {
Dispatcher.Invoke(() =>
           {
               _startButton.IsEnabled = true;
               _radioGroup.IsEnabled = true;
           });

WriteLine("--- Error received by OnConversationErrorHandler() ---");
WriteLine("Error code: {0}", e.SpeechErrorCode.ToString());
WriteLine("Error text: {0}", e.SpeechErrorText);
WriteLine();
        }

/// <summary>
/// Writes the line.
/// </summary>
voidWriteLine()
        {
WriteLine(string.Empty);
        }
/// <summary>
/// Writes the line.
/// </summary>
/// <param name="format">The format.</param>
/// <param name="args">The arguments.</param>
voidWriteLine(string format, paramsobject[] args)
        {
varformattedStr = string.Format(format, args);
Trace.WriteLine(formattedStr);
Dispatcher.Invoke(() =>
            {
```

```
                _logText.Text += (formattedStr + "\n");
                _logText.ScrollToEnd();
            });
        }

/// <summary>
/// Gets the subscription key from isolated storage.
/// </summary>
/// <returns></returns>
privatestringGetSubscriptionKeyFromIsolatedStorage()
        {
stringsubscriptionKey = null;

using                           (IsolatedStorageFileisoStore                                =
IsolatedStorageFile.GetStore(IsolatedStorageScope.User                                      |
IsolatedStorageScope.Assembly, null, null))
            {
try
                {
using                       (variStream                       =                       new
IsolatedStorageFileStream(IsolatedStorageSubscriptionKeyFileName,       FileMode.Open,
isoStore))
                    {
using (var reader = newStreamReader(iStream))
                        {
subscriptionKey = reader.ReadLine();
                        }
                    }
                }
catch (FileNotFoundException)
                {
subscriptionKey = null;
                }
            }
if (string.IsNullOrEmpty(subscriptionKey))
            {
subscriptionKey = DefaultSubscriptionKeyPromptMessage;
            }
returnsubscriptionKey;
        }

/// <summary>
/// Saves the subscription key to isolated storage.
/// </summary>
/// <param name="subscriptionKey">The subscription key.</param>
privatevoidSaveSubscriptionKeyToIsolatedStorage(stringsubscriptionKey)
        {
```

```
using (IsolatedStorageFileisoStore =
IsolatedStorageFile.GetStore(IsolatedStorageScope.User |
IsolatedStorageScope.Assembly, null, null))
            {
using (varoStream = new
IsolatedStorageFileStream(IsolatedStorageSubscriptionKeyFileName, FileMode.Create,
isoStore))
                {
using (var writer = newStreamWriter(oStream))
                    {
writer.WriteLine(subscriptionKey);
                    }
                }
            }
        }

/// <summary>
/// Handles the Click event of the subscription key save button.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="RoutedEventArgs"/> instance containing the event
data.</param>
privatevoidSaveKey_Click(object sender, RoutedEventArgs e)
        {
try
            {
SaveSubscriptionKeyToIsolatedStorage(SubscriptionKey);
MessageBox.Show("Subscription key is saved in your disk.\nYou do not need to paste the
key next time.", "Subscription Key");
            }
catch (System.Exception exception)
            {
MessageBox.Show("Fail to save subscription key. Error message: " + exception.Message,
"Subscription Key", MessageBoxButton.OK, MessageBoxImage.Error);
            }
        }

privatevoidDeleteKey_Click(object sender, RoutedEventArgs e)
        {
try
            {
SubscriptionKey = DefaultSubscriptionKeyPromptMessage;
SaveSubscriptionKeyToIsolatedStorage("");
MessageBox.Show("Subscription key is deleted from your disk.", "Subscription Key");
            }
catch (System.Exception exception)
            {
```

```
MessageBox.Show("Fail to delete subscription key. Error message: " +
exception.Message,
"Subscription Key", MessageBoxButton.OK, MessageBoxImage.Error);
            }
        }

/// <summary>
/// Helper function for INotifyPropertyChanged interface
/// </summary>
/// <typeparam name="T">Property type</typeparam>
/// <param name="caller">Property name</param>
privatevoidOnPropertyChanged<T>([CallerMemberName]string caller = null)
        {
var handler = PropertyChanged;
if (handler != null)
            {
                handler(this, newPropertyChangedEventArgs(caller));
            }
        }

privatevoidRadioButton_Click(object sender, RoutedEventArgs e)
        {
// Reset everything
if (_dataClient != null)
{
                _dataClient.Dispose();
            }
            _dataClient = null;

            _logText.Text = "";
_startButton.IsEnabled = true;
            _radioGroup.IsEnabled = true;
}
    }
}
```

Apéndice B.
Test para las pruebas

A. Audio Limpio:

Se trata de un audio completamente limpio de ruido de fondo, estructurado y con un solo hablante. Es el audio ideal.

B. Audio de Internet:

Se trata de un audio capturado directamente de Internet. En concreto hemos escogido cuentos digitales. Que son audios sin mucho ruido de fondo, estructura y con un solo hablante. También hemos usado podcast de Radio Nacional de España.

C. Audio con Calidad Radio FM:

Se trata de un audio que hemos grabado de un radio en directo.

Bibliografía

[1] Chen, S. Beeferman, D. Rosenfeld, R. Evaluation Metrics for Language Models. En: http://www.ee.columbia.edu/~stanchen/papers/d019c.pdf, bajado el 13 de febrero de 2016

[2] CommunityTool Box (2016) Capítulo 3, sección 14: SWOT Analysis: Strengths, Weaknesses, Opportunities, and Threats En: http://ctb.ku.edu/en/table-of-contents/assessment/assessing-community-needs-and-resources/swot-analysis/main Consultado el 15 de febrero de 2016

[3] Cruz Rodríguez, JM, Brito Santana J (2014) Sistema Informático de Soporte al Análisis del Discurso (SISAD). Herramienta para el estudio de la identidad en el discurso ordinario y literario, I Symposium Internacional estudios sobre discurso y sociedad, Sevilla.

[4] D. H. Bailey and P. Swarztrauber. The fractional Fourier transform and applications.*SIAM Rev.*, 33(3):389–404, 1991.

[5] Fernández, J (2016). En Hispanoteca. Lengua y cultura. Consultado el 6 de febrero de 2016, en http://www.hispanoteca.eu/

[6] Fitzpatrick, D (2007) Sistema Auditivo En: Purves, D. Agustine, G.J. Fitzpatrick, D. Hall, W.C. Lamantia, A-S. Macnamara J.O. Williams, S.M. (Eds.) Neurociencia (pp. 309-343) Madrid: Editorial Médica Patamericana S.A.

[7] Ghai, W. Singh, N. (2012) Literature Review on Automatic Speech Recognition. International Journal of Computer Applications, 41(8) 42-50.

[8] Huang, X. Baker, J. & Reddy R (2014). A Historical Perspective of Speech Recognition. Communications of the ACM, 57 (1), 94-103.

[9] Instituto Cervantes. Consultada el 7 de junio de 2016. http://cvc.cervantes.es/ensenanza/biblioteca_ele/diccio_ele/diccionario/analisisdiscurso.htm

[10] Jurafsky, D y Manning, C. (2016). Natural LanguajeProcessing. Evaluation and Perplexity En: https://class.coursera.org/nlp/lecture, consultado el 13 de febrero de 2016

[11] Jurafsky,D, Martin, J.H. (Ed.) (2012) Speech and Language Processing: An Introduction to Natural language Processing, Computational Linguistics, and Speech Recognition Editorial Cram 101, Inc (2011)

[12] Kacur, J and Rozinaj, G. (200) Practical Issues of Building Robust HMM Models Using HTK and SPHINX Systems En: http://www.intechopen.com/books/speech_recognition/Consultado el 13 de febrero de 2016

[13] Llamas Bello,C y Cardeñoso Pelayo, V. (1997) Reconocimiento automático del habla. Técnicas y aplicación. Valladolid, España: Secretariado de publicaciones e intercambio científico, Universidad de Valladolid.

[14] Management Sciens for Health y UNICEF (2016) The guide to managing for quality, EN: http://erc.msh.org/quality/ittools/itswot.cfm, consultado el 15 de febrerp de 2016

[15] Real Academia de la Lengua Española-RAE (2015) Disponible en: www.rae.es/sites/default/files/Datos_generales_v.0.82.pdf Consultada: 17 de noviembre de 2015.

[16] Real Academia Española de la Lengua-RAE (2010) Ortografía de la Lengua Española. Madrid: Espasa Libros, S.L.U

[17] Real Academia Española de la Lengua-RAE (2016) Consultada el 6 de febrero de 2016, en http://dle.rae.es/?id=N7BnIFO

[18] Ríos Mestre, A (1999) La transcripción fonética automática del diccionario electrónico de formas simples flexivas del español: estudio fonológico en el léxico. En Estudios de Lingüística del Español (Volumen 4) En http://elies.rediris.es/elies4/ consultado el 6 de enero de 2016

[19] Saini, P & Kaur P. (2013). Automatic Speech Recognition: A Review. International Journal of Engineering Trends and Technology, 4(2)132-136.

[20] Sinclair, J. (1996). *Preliminary recommendations on corpus typology.* EAGLES Document EAG-TCWG-CTYP/P. Consultado en http://www.ilc.cnr.it/EAGLES96/corpustyp/corpustyp.html

[21] Sinclair, J. 2005. "Corpus and Text - Basic Principles" en *Developing Linguistic Corpora: a Guide to Good Practice*, ed. M. Wynne. Oxford: OxbowBooks: 1-16. Disponible online en http://ahds.ac.uk/linguistic-corpora/ [Acceso el 8 de febrero de 2016].

[22] Stevens, S.S. and Volkmann, J. (1940). The Relation of Pitch to Frequency: A Revised Scale. Source: The American Journal of Psychology, 53(3), 329-353.

[23] Team FME (2013) SWOT analysis. StrategySkills Descargado de: http://free-management-ebooks.com el 14 de febrero de 2016

[24] Therese, S y Lingam, C. (2013) Review of Feature Extraction in Automatic Speech Recognition. International Journal of Scientific Engineering and Technology, 2(6), 479-484.

[25] Wynne, M (editor). 2005. *Developing Linguistic Corpora: a Guide to Good Practice.* Oxford: OxbowBooks. Disponible online en http://ahds.ac.uk/linguistic-corpora/ [Acceso el 8 de febrero de 2016].