Detección de palabras claves en lenguajes sin datos de entrenamiento

Pablo Brusco[1], Luciana Ferrer[1,2], Agustín Gravano[1,2]

[1] Departamento de Computación, FCEN, UBA
[2] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
{pbrusco,lferrer,gravano}@dc.uba.ar

Resumen Estudiamos el problema de detección de palabras claves (*keyword-spotting*) para idiomas que no disponen de corpus de datos con grabaciones y transcripciones fonéticas. Este problema es de central importancia para poder realizar búsquedas en bases de datos de grabaciones de habla. Usando el Boston University Radio Speech Corpus como corpus de referencia, analizamos diversas topologías y parametrizaciones de Modelos Ocultos de Markov para la detección de palabras sobre habla continua. Los modelos se basan en el uso de "fillers" para palabras no buscadas, y empleamos fonemas como unidades mínimas de detección. Para las pruebas, utilizamos un conjunto de 20 keywords entrenadas con 14 minutos de datos transcriptos y fillers entrenados con 7 horas sin transcripciones. Los resultados muestran que el mejor modelo alcanza rendimientos superiores a un 0.47 de FOM promedio, un porcentaje de detecciones correctas del 72.1 % y 3.95 falsas alarmas por hora por keyword.

Palabras claves: Keyword-Spotting, Automatic Speech Recognition, Hidden Markov Models, Speech Data Mining.

1. Introducción

El problema de *Keyword-Spotting* en habla (también llamado *Word-Spotting*) consiste en detectar palabras claves en grabaciones de habla continua. Este problema es de central importancia en aplicaciones de búsqueda de información en bases de datos de habla crudas (es decir, que no cuentan con transcripciones ni otras anotaciones).

El Keyword-Spotting puede ser considerado un sub-problema del Reconocimiento Automático del Habla (ASR, del inglés *Automatic Speech Recognition*) y, por lo tanto, utilizando los mismos algoritmos podemos transcribir a palabras todo el audio donde deseamos ubicar las keywords, y luego buscar texto en texto como es habitual realizar búsquedas. De todas formas, este mecanismo tiene una gran desventaja. El ASR no es un problema fácil de resolver y solo funciona bien bajo una gran cantidad de supuestos. Entre ellos, se necesitan grandes corpus de datos de entrenamiento (es decir, habla transcripta a nivel de palabras e incluso, para mejores resultados, a nivel fonemas) que suelen escasear en la mayoría de

los lenguajes. La gran mayoría de los trabajos (papers, tesis, aplicaciones comerciales, etc.) se basan en tener cierto conocimiento del idioma. Por ejemplo, suelen utilizar corpus como DARPA Resource Management Database o TIMIT para reconocer palabras en inglés.

En este trabajo analizamos la factibilidad de construir un sistema bajo la restricción de tener una cantidad mínima de audio transcripto para entrenamiento. Una vez construido, exploramos distintas dimensiones en busca de una performance aceptable para este tipo de tareas.

2. Trabajo Previo

Los antecedentes del problema de detección de palabras claves están estrechamente relacionados con los antecedentes en reconocimiento del habla general. Desde la década de los 70, trabajos como el de Bridle [1] se basaron en el uso de *Dynamic Time Warping* (DTW). DTW es una técnica que utiliza programación dinámica para medir similitudes entre dos secuencias temporales que pueden tener desfasajes, en este caso, dos secuencias de audio en búsqueda de coincidencias. Esta técnica probó ser útil cuando el reconocimiento se hace sobre datos parecidos a los de entrenamiento; pero en cuanto se agrega variabilidad, demostró no escalar. En especial para la detección de palabras claves, importantes aportes fueron presentados por Bridle, quien introduce la utilización de modelos de *filler*: representación de palabras fuera del conjunto de palabras buscadas.

Más recientemente, a partir del trabajo de Rabiner [9], se ha popularizado el uso de Modelos Ocultos de Markov (HMMs) de distribuciones continuas utilizando palabras o sub-palabras como unidades acústicas para la tarea de reconocimiento del habla, desde entonces, la técnica más utilizada. El uso de sub-palabras facilita el reuso de entrenamiento entre palabras que comparten ciertas partes permitiendo, por ejemplo, el reconocimiento de palabras fuera del conjunto de entrenamiento.

En el caso de la detección de palabras claves, el uso de estos modelos fue aplicado por primera vez por Rohlicek y col. [10] y Rose y Paul [11] marcando el comienzo de numerosos estudios basados en sus ideas. Cabe destacar que el trabajo de Rose y Paul [11] indica que para conseguir una buena performance se necesita entrenar un modelo de filler complejo utilizando grandes cantidades de datos transcriptos o modelos preexistentes. En particular, utilizan un HMM que conecta completamente todos los posibles modelos de sub-unidades del lenguaje. Por otro lado, hay trabajos como el de Weintraub [13] que complejizan aún más estos modelos utilizando modelos de vocabulario completo que excluyen a las keywords.

Esta técnica funciona muy bien pero requiere una gran cantidad de datos de entrenamiento. Trabajos como el de Moore [7] muestran que un sistema de reconocimiento general disminuye hasta un 15 % de efectividad al trabajar con poca cantidad de datos (usando bases de datos de menos de 100 horas).

El problema de detección de palabras claves tiene ciertas características que hacen posible realizar un sistema con corpus mucho menores a los necesarios

para el reconocimiento general de habla, manteniendo una performance razonable. Existen trabajos basados en la idea de utilizar la menor cantidad posible de datos. Por ejemplo, Thambiratnam [12] dedica un capítulo de sus tesis de doctorado a evaluar sistemas para lenguajes distintos al inglés. En particular, muestra cómo se ve afectado un detector de palabras claves al variar entre 164, 15 y 4 horas de entrenamiento en inglés, español e indonés. Su trabajo concluye que el reconocimiento de palabras claves sufre menos en performance ante la falta de datos que las tareas de reconocimiento general del habla. De todas maneras, este trabajo utiliza la totalidad de horas de entrenamiento para formar modelos de fonemas reuniendo información de todas las palabras presentes en las grabaciones.

Más allá de compartir las mismas motivaciones que el trabajo de Thambiratnam [12], la forma en que encaramos la solución al problema es distinta. Basamos esta afirmación en el hecho de utilizar 14 minutos de datos transcriptos y el resto de los datos (no transcriptos) para el entrenamiento de los modelos de fillers. De esta manera el trabajo previo a la construcción del reconocedor disminuye notablemente.

También, el artículo de Garcia y Gish [4] explica que con solo 15 minutos de transcripciones junto a un modelo de reconocimiento que define sus propias unidades de sonido en base a modelos segmentales se puede alcanzar una gran performance. De todas maneras, en esta tesis decidimos utilizar el enfoque estándar de HMMs con distribuciones continuas que han probado funcionar muy bien con grandes cantidades de datos.

3. Diseño del Sistema

Para nuestros experimentos construimos un sistema usando la herramienta Hidden Markov Model Toolkit (HTK), la cual integra funciones útiles para trabajar con HMMs [14]. Nuestro sistema se basa en la gramática de estados finitos representada en la Figura 1, que refleja los posibles estados en los que puede estar el proceso de reconocimiento sobre una grabación. En este caso, se permite una cantidad indeterminada de apariciones de keywords y fillers, intercaladas libremente.

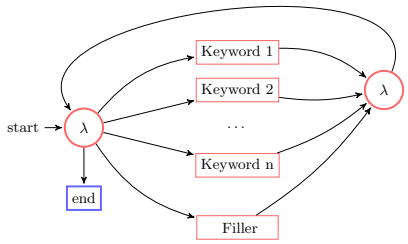


Figura 1: Gramática de estados finitos

Cada keyword se encuentra modelada por su secuencia de fonemas y, a su vez, dichos fonemas son modelados por HMMs con topología *left-right* de tres estados emisores.[1] Por su parte, todos los fillers se representan mediante un único HMM ergódico de tres o cinco estados emisores.[2] Cada uno de estos estados tiene asociada una probabilidad de emisión modelada mediante *Mezclas de Gaussianas* con una cantidad determinada de mezclas. Tanto la cantidad de mezclas como la cantidad de estados del HMM ergódico de los fillers serán estudiados en los experimentos descriptos en la Sección 5.

Por último, una vez entrenado un HMM como el detallado, lo ponemos en funcionamiento de la siguiente manera. Dada una grabación nueva en la cual deseamos detectar la aparición de keywords, utilizamos el algoritmo de decodificación de *Viterbi* para encontrar el camino óptimo en el HMM (es decir, el camino de mayor verosimilitud dada la grabación nueva). De esta manera, el sistema devuelve la secuencia más probable de palabras y fillers, y devuelve además un puntaje de verosimilitud que utilizaremos para las métricas en las evaluaciones.

4. Corpus de Datos

El corpus de datos utilizado es el Boston University Radio Speech Corpus [8]. Consiste en grabaciones radiales de lecturas de noticias hechas por locutores profesionales. Consta de siete lectores (cuatro hombres y tres mujeres) dedicados a anunciar noticias en radio FM, que produjeron grabaciones en dos etapas: por un lado, en un laboratorio de la Universidad de Boston donde se les pidió que leyeran 24 historias alternando su manera de hablar entre forma radial y no radial; por otro lado, más de siete horas de audio tomadas directamente del programa radial en los estudios de la WBUR (una radio local de Boston). El audio fue digitalizado a 16kHz y 16 bits.

Tanto las historias leídas como los programas de radio fueron anotados con transcripciones ortográficas generadas a mano con precisión a nivel palabra. Es decir, cada palabra está marcada con un tiempo que indica el momento en que terminó de decirse la misma en la grabación. Luego, se estima el comienzo de la palabra siguiente tomando el fin de la inmediata anterior.

Analizando el tamaño y la escasa cantidad de locutores, se tomó la decisión de dividir al corpus en seis grupos uniformes. Para ello, se listaron todos los archivos de grabaciones ordenados alfabéticamente y se recorrieron asignando cada uno a un grupo de pertenencia, entre 1 y 6. De esta manera, la cantidad de grabaciones por grupo pertenecientes a cada hablante y en cada situación es pareja. Una vez separados, se tomaron 4/6 del total (7.31 horas) de las frases para entrenamiento, 1/6 (1.8 horas) para testeo y 1/6 (1.9 horas) para resultados finales que se ignoran durante todo el desarrollo del sistema. Junto con esta

[1] En una topología *left-right*, los estados se disponen de izquierda a derecha, de modo que solamente se puede avanzar hacia la derecha, sin volver atrás.

[2] En una topología *ergódica* se permite pasar de cualquier estado a cualquier otro.

división, se adjuntaron a cada grabación las transcripciones correspondientes que permiten realizar el entrenamiento y la evaluación.

También se realizó una división del corpus por hablante para una de las pruebas finales. En este caso se dividió el corpus por hablante y se hizo una rotación en la cual se selecciona de a un hablante y luego se toman todos sus archivos para testeo y el resto del corpus para entrenamiento. De esta manera se hace un estudio del sistema cuando el hablante no es parte del corpus de entrenamiento. Una vez fijada la porción del corpus para entrenamiento, se procedió a marcar las ocurrencias de keywords en el audio. De esta manera, los datos de entrenamiento quedaron divididos en dos partes. La primera, 14 minutos de datos transcriptos para 20 keywords. La segunda, el resto del tiempo (7 hs aproximadamente) fue dividido en intervalos regulares para el entrenamiento de fillers.

El tamaño que se utiliza para cada uno de los intervalos regulares fue un factor a estudiar. Analizaremos en la sección de resultados como varía el modelo de fillers a medida que este intervalo cambia entre 100 y 500 ms.

Además, decidimos trabajar con un conjunto de 20 keywords con nueve o más letras seleccionadas por ser las más frecuentes del corpus. En la Tabla 1 puede verse el conjunto de keywords seleccionadas para las pruebas. Las columnas relacionadas con apariciones muestran la cantidad de veces que aparece cada keyword en las transcripciones de referencia en entrenamiento, testing y control respectivamente. Por otro lado, se muestra la duración promedio de las palabras en el corpus.

5. Evaluación

Durante la evaluación buscamos medir la performance del sistema a desarrollar, en busca de analizar su factibilidad en la práctica. Para ello simulamos el sistema bajo diversas parametrizaciones. Entrenamos los modelos usando datos transcriptos para entrenamiento, y luego testeamos sobre datos de referencia.

En nuestros experimentos estudiamos varios de los factores antes mencionados, en busca de optimizarlos para alcanzar el mejor rendimiento posible del sistema. Las características principales en las que se concentró el estudio fueron: 1) la cantidad de mezclas Gaussianas para los modelos; 2) el intervalo de relleno con fillers; y 3) la cantidad de estados en el modelo de filler.

El testeo se realizó sobre 1.8 horas de grabaciones en donde ocurren 313 apariciones de las keywords elegidas. Una vez que finalizó una prueba, el resultado consiste en un archivo que contiene, para cada grabación, detecciones de todas las keywords con tiempo de inicio y fin junto con un puntaje (una probabilidad) otorgado por el algoritmo de *Viterbi*. Estos resultados fueron luego comparados con las transcripciones de referencia, para estimar la performance del sistema.

La manera elegida para estimar dicha performance consiste en contabilizar la cantidad de aciertos (Hits) y falsas alarmas (FA) que se encuentran en el resultado. Diremos que una detección res es un Hit si existe una aparición de la keyword en la referencia (que llamamos ref) tal que las palabras son idénticas y el tiempo medio de ref está entre los tiempos inicial y final de la detección res.

Es decir, $t_i(res) < t_m(ref) < t_f(res)$, donde t_i, t_m y t_f representan el tiempo inicial, medio y final de una aparición. De otra manera, esa detección constituye una falsa alarma.

Luego, calculamos la *curva ROC* (*Receiver Operating Characteristic*), una medición estándar muy utilizada que grafica la tasa de Hits contra el número de falsas alarmas por keyword por hora (fa/kw-hr). Para reducir esta representación a un único número, utilizamos la medida *FOM* (*Figure of Merit*) que equivale a la probabilidad promedio de detecciones correctas de 1 a 10 falsas alarmas por hora (ver Manos [6] y Brusco [2] para más detalles). Para mostrar los resultados en forma compacta, usamos el FOM promedio sobre todas las keywords analizadas.

5.1. Evaluación de tres características del sistema

1) Cantidad de mezclas Gaussianas: Las mezclas Gaussianas determinan las probabilidades de observación a partir de cada estado, y en consecuencia brindan flexibilidad en cuanto a reconocer mayor o menor variabilidad a partir de vectores acústicos. Es de esperar que modelos con mayor cantidad de mezclas reconozcan con más precisión sonidos similares a los utilizados para el entrenamiento. De todas maneras, un problema que puede surgir es el de sobre-ajuste (*overfitting*) con respecto a estos datos.

La Figura 2 resume los experimentos realizados, los cuales se describen en las próximas secciones. Además, en los gráficos (a), (b) y (c) de dicha figura, el eje x corresponde a la cantidad de mezclas Gaussianas empleada. Con esto buscamos estudiar la evolución de la performance de cada configuración del sistema en función de dicha cantidad. En estos experimentos, aumentamos cada cinco reestimaciones el número de mezclas Gaussianas por estado, de a dos por incremento, y variando entre una y al menos 65 componentes.

2) Duración del intervalo para fillers: En nuestros experimentos variamos la duración del intervalo con el que se marcan los fillers, en busca de la duración óptima. Experimentamos con duraciones de 100, 200, 350 y 500 milisegundos, teniendo en cuenta que 200 ms se aproxima a la duración promedio de una sílaba en el idioma inglés [3, 5] y 500 ms a la de una palabra completa (la duración promedio de las palabras en este corpus es 480 ms).

Puede observarse en la Figura 2(a) la notable superioridad en el FOM promedio cuando usamos intervalo de 200 ms, especialmente a partir de las 25 mezclas. En cambio, los resultados obtenidos para 500 ms no mejoran al aumentar las mezclas de Gaussianas. Por otro lado, los resultados obtenidos para 100 y 350 ms, aunque lejos del óptimo, muestran una tendencia a acercarse a un FOM de 0.15. En consecuencia, en las siguientes pruebas trabajamos con 200 ms, para seguir explorando otras dimensiones de análisis.

3) Cantidad de estados en los fillers: A diferencia de los fonemas de keywords, que son modelados usando tres estados emisores, para el filler variamos entre tres y cinco estados, para estudiar la performance del sistema en cada caso. Nuestra intuición a priori nos indicaba que con cinco estados se lograría mayor

flexibilidad en los modelos y, por lo tanto, menor cantidad de falsas alarmas. Mediante este experimento buscamos validar o refutar dicha intuición.

La Figura 2(b) resume los resultados obtenidos de comparar tres contra cinco estados, variando al mismo tiempo la cantidad de mezclas, con modelos entrenados utilizando fillers cada 200 milisegundos (según lo discutido en la sección anterior). Observamos una rápida adaptación del modelo de cinco estados y un leve progreso en el de tres con una marcada superioridad en especial de 30 a 40 mezclas Gaussianas. En consecuencia, concluimos que conviene trabajar con cinco estados. Dada la marcada diferencia de performance entre tres y cinco estados, queda para trabajo futuro experimentar con un número mayor de estados.

En conclusión, de este primer estudio obtenemos una clase de modelos que maximiza nuestras mediciones. Las características óptimas del sistema son: 1) 37 mezclas de Gaussianas; 2) intervalo para completar con fillers de 200 ms de longitud; y 3) cinco estados en el modelo de los fillers.

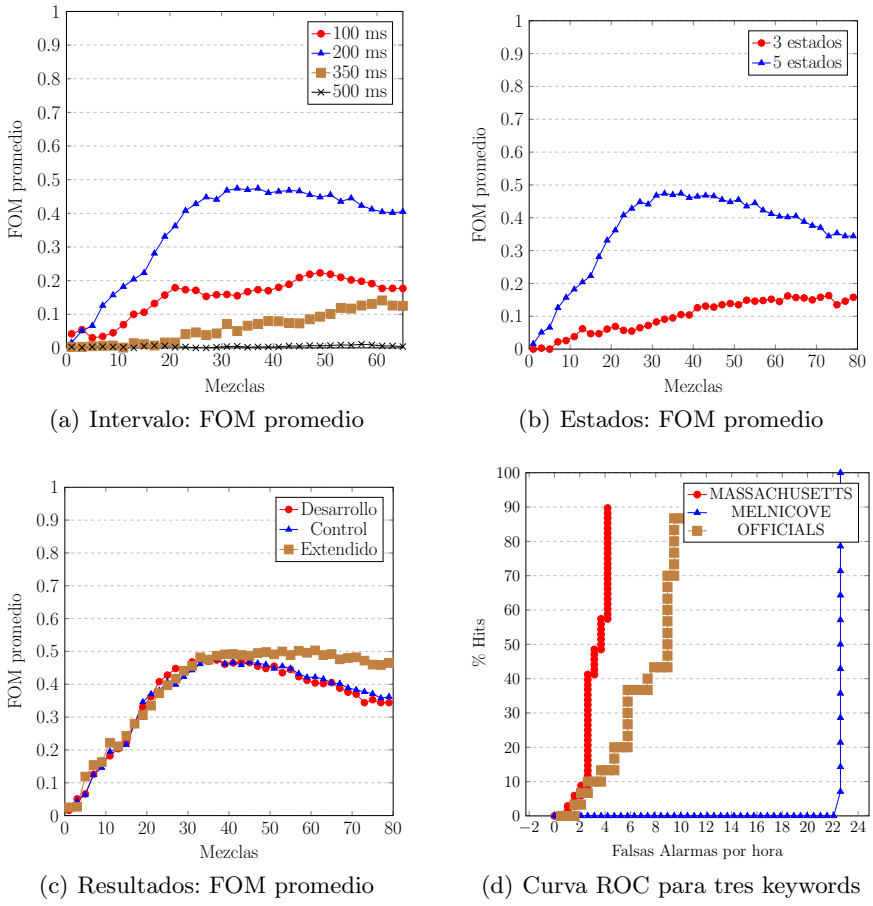
5.2. Prueba sobre set de control

Para las siguientes pruebas se utilizó la sexta parte del corpus reservada para este propósito (ver Sección 3). Utilizando los mejores modelos entrenados para las pruebas en desarrollo, se testeó sobre esta reserva buscando obtener resultados más realistas por tratarse de datos no utilizados hasta el momento.

	Keyword	DP	AE	AT	AC	%H	FA	FOM	S	I
1	ACCORDING	0.503	50	20	21	28.6	01	0.278	0	01
2	ADMINISTRATION	0.750	67	12	07	85.7	08	0.572	0	08
3	ASSOCIATION	0.748	29	07	14	50.0	02	0.459	0	02
4	COMMITTEE	0.391	57	11	10	70.0	08	0.663	0	08
5	DEMOCRATIC	0.598	46	14	19	52.6	02	0.499	0	02
6	DEMOCRATS	0.650	32	10	06	100	16	0.518	6	10
7	EDUCATION	0.639	37	12	15	73.3	11	0.408	0	11
8	ENVIRONMENTAL	0.702	47	11	15	60.0	01	0.568	0	01
9	GOVERNMENT	0.460	58	22	15	26.7	01	0.263	0	01
10	LAWMAKERS	0.682	36	02	06	83.3	04	0.702	0	04
11	LEGISLATURE	0.692	37	10	11	81.8	05	0.608	0	05
12	MASSACHUSETTS	0.744	318	71	68	89.7	08	0.606	0	08
13	MELNICOVE	0.622	65	19	14	100	42	0.000	0	42
14	OFFICIALS	0.486	145	25	30	93.3	20	0.283	0	20
15	POLITICAL	0.541	66	12	11	72.7	04	0.679	0	04
16	PRESIDENT	0.512	51	13	18	88.9	09	0.629	0	09
17	REPUBLICAN	0.637	39	08	13	23.1	04	0.186	0	04
18	SPRINGFIELD	0.676	24	07	04	25.0	00	0.250	0	00
19	SUPERINTENDENT	0.728	28	10	05	60.0	00	0.600	0	00
20	YESTERDAY	0.596	99	17	24	87.5	04	0.763	0	04
	Resultados generales					72.1	150	0.477	6	144

Tabla 1: Resultado control por keyword (37 mezclas sobre 1.9 hs de grabaciones) DP: Duración Promedio; AE: Apariciones en Entrenamiento; AT: Apariciones en Testing; AC: Apariciones en Control; %H: Porcentaje de Hits; FA: Falsa Alarmas; S: Sustituciones, I: Inserciones.

La Tabla 1 resume los resultados de esta evaluación para cada una de las 20 keywords seleccionadas. Los resultados generales del mejor modelo superan 0.47



(a) Intervalo: FOM promedio

(b) Estados: FOM promedio

(c) Resultados: FOM promedio

(d) Curva ROC para tres keywords

Figura 2: Resultados en desarrollo y control

de FOM promedio, 72.1 % de Hits y 3.95 falsas alarmas por hora por keyword (con 326 apariciones de keywords totales en 1.9 horas de grabaciones).

La Figura 2(d) muestra la curva ROC para tres keywords representativas. Puede verse en el eje x la cantidad de falsas alarmas por hora y el impacto que podría producir en el porcentaje de Hits optar por aceptar esa cantidad como umbral de detección para el sistema. En el caso de `MASSACHUSETTS` permitir 5 falsas alarmas por hora produciría un porcentaje de Hits máximo, en cambio, para `MELNICOVE` impactaría dejando un pésimo resultado.

Por otra parte, la Figura 2(c) muestra que los mejores modelos poseen un comportamiento casi idéntico cuando se los testea sobre los datos de desarrollo y sobre los datos de control (círculos rojos y triángulos azules, respectivamente). Como prueba secundaria, utilizamos la misma configuración que produjo los mejores resultados en desarrollo y entrenamos nuevos modelos sobre los 5/6 del

corpus utilizados en desarrollo (es decir, con un quinto más de datos). Llamamos a esta prueba "resultado extendido" (cuadrados marrones en la figura), y observamos un incremento en su performance, que se hace más notorio a partir de las 40 mezclas. Este último resultado es de particular importancia, pues muestra que, con más datos de entrenamiento, el sistema responde variando su performance de manera notable. Esto último hace pensar que sólo 15 minutos de datos transcriptos son insuficientes para que el sistema llegue a su mejor rendimiento.

5.3. Prueba por Hablante

Este experimento midió la performance del sistema cuando se entrena con todos los hablantes salvo uno (estrategia *leave-one-out*) y luego se evalúa sobre ese hablante. Esto lo realizamos para analizar la robustez del sistema ante hablantes desconocidos. Para ello, se fue tomando cada uno de los siete hablantes presentes en el corpus y se entrenó con el resto del grupo, luego se probó sobre el hablante elegido y se tomaron los resultados. Este proceso se repitió por cada hablante, recolectando las detecciones en una sola tabla sobre la cual se calcula la performance global.

Los resultados (medidos en FOM promedio) fueron 0.335, 0.351 y 0.253 para las mujeres y 0.034, 0.126, 0.473 y 0.126 para los hombres. La performance global fue de 0.102. En consecuencia, podemos observar una gran degradación de la performance del sistema en general, aunque para ciertos hablantes funcionó al mismo nivel que los mejores resultados obtenidos en desarrollo, como los números en el tercer hombre. Estos resultados resultan esperables, dado que entrenar con sólo seis sujetos debería ser insuficiente para generalizar a nuevos hablantes.

Para evaluar los resultados, hicimos una consulta informal a una estudiante de fonoaudiología en donde le pedimos que indique similitudes y diferencias entre las voces presentes y también características relevantes que pueda detectar. El resultado indicó que la voz de las tres mujeres es bastante similar entre sí y, además, la forma de hablar de la mujer 1 con la del hombre 3 tienen mucho en común. Por otra parte, la voz del hombre 1 parece ser muy monótona comparada con el resto. Esta información se ajusta con precisión a los resultados obtenidos.

6. Conclusiones y Trabajo Futuro

Se construyó y estudió un sistema para detectar palabras claves en habla continua con una cantidad mínima de datos de entrenamiento. Utilizamos HMMs para modelar un conjunto predefinido de keywords, y un modelo de fillers para el resto de los sonidos presentes en las grabaciones (silencio y otras palabras). Luego, utilizando el Boston University Radio Speech Corpus entrenamos y evaluamos diversos modelos que permitieron la búsqueda de la mejor configuración. Las características exploradas son la cantidad de mezclas Gaussianas para los modelos (la cantidad óptima rondó las 37), la longitud del intervalo de relleno con fillers (200 ms), y la cantidad de estados en el modelo de filler (cinco estados).

En base a nuestro estudio, podemos concluir que es posible realizar un sistema con las características planteadas con un rendimiento aceptable y mejorable, sin la necesidad de incluir técnicas muy complejas ni grandes cantidades de datos transcriptos. Para poder construir un sistema con las características que hemos estudiado, se necesita un conjunto de grabaciones sin transcripciones (siete horas o más) con buena calidad de audio. Además, se deben anotar en los audios la ubicación exacta de cada keyword, y la suma de los tiempos de todas las keywords debe rondar los 14 minutos (asumiendo 20 keywords).

Por otra parte, este estudio deja abierta una serie de preguntas sin responder relacionadas principalmente con la búsqueda de mejores resultados. Alguno de los caminos posibles a seguir son: ver cómo afecta al sistema la incorporación de nuevos datos de entrenamiento, ya sea mediante más repeticiones o mediante nuevas keywords; desarrollar una implementación de modelos de fillers más complejos, manteniendo la restricción de no tener datos transcriptos; estudiar el impacto de variar factores como pesos en la gramática o penalidad de inserción de palabras en el algoritmo de Viterbi; y considerar técnicas para lograr que el sistema sea más resistente a cambios de hablante.

Referencias

[1] John S Bridle. "An efficient elastic-template method for detecting given words in running speech". En: *Brit. Acoust. Soc. Meeting*. 1973.

[2] Pablo Brusco. "Keyword-spotting en idiomas sin datos de entrenamiento". Tesis de lic. Depto. de Computación, FCEN, UBA, Buenos Aires, Argentina, 2014.

[3] W Nick Campbell. "Syllable-based segmental duration". En: *Talking machines: Theories, models, and designs* (1992), págs. 211-224.

[4] Alvin Garcia y Herbert Gish. "Keyword spotting of arbitrary words using minimal speech resources". En: *ICASSP*. Vol. 1. 2006.

[5] Raymond D Kent y LL Forner. "Speech segment duration in sentence recitations by children and adults." En: *Journal of Phonetics* (1980).

[6] Alexandros Sterios Manos. "A study on out-of-vocabulary word modelling for a segment-based keyword spotting system". Tesis doct. MIT, 1996.

[7] Roger K Moore. "A comparison of the data requirements of automatic speech recognition systems and human listeners." En: *INTERSPEECH*. 2003.

[8] Mari Ostendorf, PJ Price y Stefanie Shattuck-Hufnagel. "The Boston University radio news corpus". En: *Linguistic Data Consortium* (1995).

[9] Lawrence Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". En: *Proceedings of the IEEE* 77.2 (1989).

[10] J Robin Rohlicek y col. "Continuous hidden Markov modeling for speaker-independent word spotting". En: *ICASSP*. IEEE. 1989.

[11] Richard C Rose y Douglas B Paul. "A hidden Markov model based keyword recognition system". En: *ICASSP*. IEEE. 1990.

[12] Albert JK Thambiratnam. "Acoustic keyword spotting in speech with applications to data mining". Tesis doct. Queensland University of Technology, 2005.

[13] Mitchel Weintraub. "Keyword-spotting using SRI's DECIPHER large-vocabulary speech-recognition system". En: *ICASSP*. Vol. 2. 1993.

[14] S Young y col. "The HTK-Book 3.4". En: *Cambridge University, Cambridge, England* (2006).