

Optimización de un Sistema de Reconocimiento de Idioma Basado en la Representación Vectorial de Fonemas

Optimising a Language Recognition System Through Phoneme-Based Vector Representation

Francisco Charro, Marco Herrera, Nataly Pozo y Andrés Rosales

Resumen — En este artículo se analiza la representación vectorial de fonemas como una alternativa para mejorar un sistema de reconocimiento de idioma (LID). Se estudian las arquitecturas, CBOW (Continuous Bag-of-Words) y Skip-gram propuestas por Mikolov. Estos modelos permiten predecir palabras dentro de un contexto mediante la generación de vectores n-dimensionales. En este trabajo se analizará la aplicación de estos modelos en unidades fonéticas más pequeñas o n-gramas.

Palabras Clave – Representación Vectorial, Reconocimiento de Idioma, Skip-gram, n-gramas, embeddings.

Abstract - This article analyzes vector representation of phonemes as an alternative to improve a language identification system (LID). CBOW (Continuous Bag-of-Words) and Skip-gram architectures proposed by Mikolov are studied. These models allow predicting words within a context by generating n-dimensional vectors. In this work we will analyze the application of these models in smaller phonetic units or n-grams.

Keywords - Vector Representation, Language Recognition, Skip-gram, n-grams, embeddings.

I. INTRODUCCIÓN

El reconocimiento de idioma es una tarea implícita en la mayoría de sistemas informáticos que interactúan con los seres humanos, sin embargo el mejoramiento de su rendimiento sigue en desarrollo. Se han probado varios enfoques, los más básicos desde la semántica formal con modelos simbólicos que siguen reglas fijas, tienen un procesamiento rápido pero ignoran las sutilezas del lenguaje y el contexto que al no ser considerados resulta en un sinnúmero de extensiones y alternativas [1]. Una opción válida a esta técnica es la semántica distribucional, que convierte el lenguaje en un modelo matemático, cambiando de un enfoque simbólico a uno estadístico, y de esta manera se pueden estudiar las relaciones

entre distribución y significación aumentando la probabilidad de éxito [2]. Los modelos estadísticos actuales basan su función de distribución en la representación vectorial de palabras, definen la distancia entre ellas en un espacio vectorial, y construyen un modelo de predicción en el que se puede pronosticar la siguiente palabra que va a aparecer en base a la palabra actual y el contexto en el que se circunda [1].

La aportación de este documento está dirigido a estudiar los efectos de la representación vectorial a nivel fonético o poli-fonético (varios fonemas unidos en una sola unidad) en la identificación de idioma.

El sistema LID implementado se presenta en la Figura 1. Consta de 2 etapas principales, el Front-End donde se extraen los parámetros principales de cada idioma, en este caso la representación vectorial fonética, y el Back-End que modela los idiomas a reconocer con técnicas basadas en i-Vectores. Estos procesos se los realiza tanto para los datos de entrenamiento como para los evaluados. Utiliza un clasificador basado en regresión logística y máxima verosimilitud estadística.

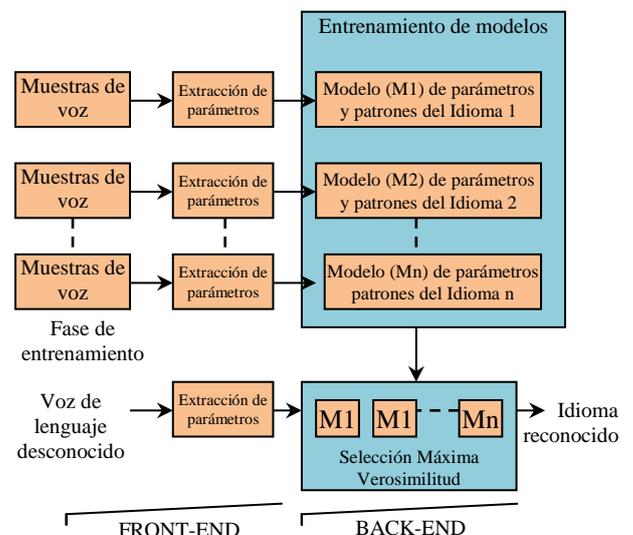


Figura 1. Diagrama básico de un sistema LID

Los autores pertenecen al Departamento de Automatización y Control Industrial de la Escuela Politécnica Nacional, Ladrón de Guevara, E11-253, Quito, Ecuador. (e-mail: {francisco.charro, marco.herrera, nataly.posov, andres.rosales@epn.edu.ec})

La técnica de modelado con la que se comprobó la calidad de la representación vectorial es conocida como PRLM (modelado de idioma con reconocimiento fonético) en paralelo o (PPRLM) [3], implementada en investigaciones anteriores por el Grupo de Tecnología del Habla (GTH) de la Universidad Politécnica de Madrid (UPM) a la que se tuvo acceso para la experimentación. Este sistema permite también modelar la frecuencia de ocurrencia de las secuencias de fonemas en cada uno de los idiomas a reconocer lo que permite centrarse en la representación vectorial únicamente [4].

Los modelos se crean a partir de las secuencias de fonemas generadas con reconocedores existentes y así se obtiene una secuencia diferente para cada reconocedor. Por lo tanto la calidad y tamaño de la base de datos es un factor importante.

El presente trabajo se ha verificado sobre la base de datos KALAKA-3 desarrollada para la competición europea de reconocimiento de idioma Albayzin LRE 2012 [5] en la condición plenty-closed, es decir se poseen datos de audio etiquetados en 6 idiomas (vasco, catalán, inglés, gallego, portugués y español) para el entrenamiento (plenty) y no admite otros diferentes para la evaluación (closed). Se tienen además 3 reconocedores de fonemas que utilizan Modelos Ocultos de Markov (HMM's) (húngaro, ruso y checo) que generan un conjunto de secuencias de fonemas para cada uno de los 6 idiomas (en el presente caso es posible obtener, 3 HMM's \times 6 idiomas = 18 distribuciones de idioma o PRLM's). Durante el reconocimiento, la muestra a ser evaluada es procesada por cada PRLM y sus salidas son promediadas en el dominio logarítmico como sistemas independientes, la distribución de los datos a evaluar se compara con los modelos y se generan scores basados en las verosimilitudes.

II. DESCRIPCIÓN DEL SISTEMA

El sistema implementado está basado en un sistema PPRLM donde se tienen seis modelos correspondientes a los seis idiomas de la base de datos KALAKA-3 por cada reconocedor de fonemas disponible o HMM (Húngaro, Ruso o Checo).

En la Figura 2 se muestra el modelo planteado, tomando como ejemplo el HMM de Húngaro y el idioma Vasco. Este sistema corresponde al que presentó mejores métricas, pero es necesario explicar las pruebas previas hasta alcanzar este modelo propuesto. A partir de ahora, llamaremos "embeddings" [6] a la representación vectorial de los fonemas.

A. FRONT-END

a. Creación de las listas de entrenamiento

Las listas de entrenamiento se crean a partir de los datos pre-adquiridos por los reconocedores fonéticos disponibles de KALAKA-3. La lista de entrenamiento consta de 4656 archivos y se los separa en dos bloques, el bloque inferior (train_DOWN, con menor cantidad de elementos) para la generación de los embeddings y el bloque superior (train_UP) se utilizará en la creación de los i-Vectores de entrenamiento de cada modelo PRLM, esta división es necesaria para evitar el sobrentrenamiento del sistema. Las listas empleadas para desarrollo constan de 458 elementos y las de evaluación de 941 elementos.

Los parámetros a definir en esta etapa son: 1) el porcentaje de cada bloque superior e inferior del entrenamiento y 2) el

número de n-gramas en los que se agruparán los fonemas de cada archivo de audio.

b. Entrenamiento de los embeddings

Se utilizan las listas de entrenamiento "train_DOWN" para la generación de embeddings a través del toolkit Wor2Vec propuesto por Mikolov [7]. Este software genera una representación vectorial de un poli-fonema utilizando el modelo CBOW o Skip-gram. En esta experimentación se probaron las dos configuraciones, y se obtuvieron vectores para bi-gramas, tri-gramas o cuatri-gramas hasta obtener la mejor "calidad" de los vectores.

Generamos una nueva lista de todos los n-gramas contenidos en las listas train_DOWN (en total seis listas, una por idioma), se cuenta el número de apariciones de los n-gramas presentes, se ordenan por frecuencia de aparición y se calcula para cada uno, un vector de la dimensión requerida según el modelo seleccionado (CBOW/Skip-Gram) y sus parámetros.

Estas listas contienen todo el vocabulario de fonemas para cada uno de los 6 idiomas, representados vectorialmente para un HMM determinado.

a. Representación vectorial de fonemas o embeddings

Para generar las listas de la secuencia de n-gramas de cada archivo de entrenamiento (train_UP), desarrollo y evaluación, se realiza un mapeo de estos archivos con la lista de embeddings para construir los pares fonema-vector que son los que nos interesan. Si un fonema no aparece en la lista de embeddings se le asigna un vector "missing grams" de valor cero. Al final se tendrán todos los archivos de entrenamiento, desarrollo y evaluación mapeados con la lista de embeddings.

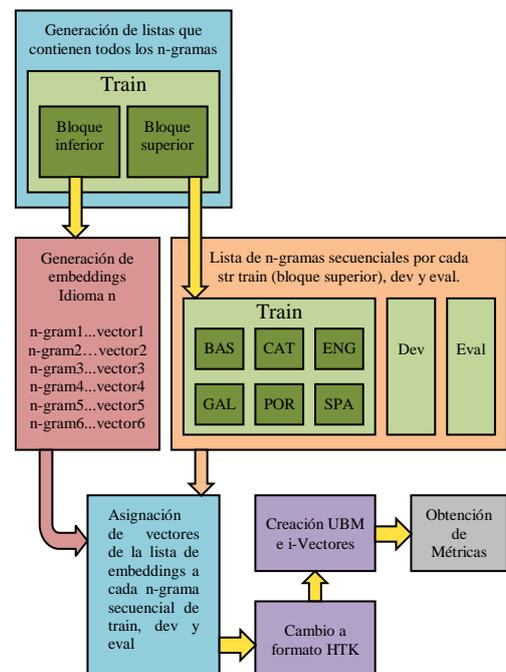


Figura 2. Diagrama de bloques del sistema LID implementado

A estas nuevas listas que contienen la información de la secuencia de n -gramas y sus embeddings se agrega información de contexto recalculando su valor con información de sus vecinos. En esta nueva y definitiva representación los vectores se calculan ponderando con un 50% del valor del vector actual, un 25% de su vector anterior y un 25% del vector siguiente. El tiempo de cálculo aumenta pero las métricas mejoran considerablemente.

Para la generación de i -Vectores mediante el fusionador disponible, se requiere cambiar los datos a formato binario HTK mediante la función de Matlab `writetk.m`. El contenido de estos archivos de extensión `.fea` es el mismo que el de las listas que contienen los vectores con contexto.

B. BACK-END

La experimentación contempla la creación de modelos independientes de idiomas o UBM (Universal Background Model) que son modelos de mezclas gaussianas (GMM) entrenados con una gran cantidad de datos. Se crearon UBM's para cada uno de los 6 idiomas a reconocer utilizando uno de los 3 HMMs disponibles, se modifican los parámetros de los algoritmos CBOW o Skip-gram hasta alcanzar los valores óptimos. En la experimentación final se crearon los seis sistemas correspondientes a cada idioma.

El UBM genera un modelo de Gaussianas y luego se entrenan los i -Vectores que representan la información de los ficheros de entrenamiento, desarrollo y evaluación. A partir de estos i -Vectores se obtienen las métricas en el sistema de fusión.

III. RESULTADOS DE LAS PRUEBAS

Las métricas utilizadas para evaluar el rendimiento del sistema son: la Función de Costo de Detección (Fact), y la Función de Costo de Decisión Promedio (Cavg) que miden el costo de decisión provocado por las tasas de error de falsa aceptación y de falso rechazo generadas en la evaluación [8].

La experimentación partió de un sistema inicial al que se le han modificado sus parámetros uno a uno hasta encontrar una configuración óptima.

Este sistema inicial tiene 2 tipos de parámetros, los propios del sistema LID y los correspondientes a la generación de los embeddings.

A. SISTEMA INICIAL

En el sistema inicial se utilizó un solo UBM del HMM de Húngaro [9] con los vectores de parámetros de los seis idiomas para entrenar los i -Vectores. En primer lugar se crearon las listas para los 6 idiomas, que contienen todos los poli-fonemas entrenados y su representación vectorial ordenados por frecuencia de aparición y de dimensión N establecida manualmente. Seguidamente se generaron las listas de poli-fonemas ordenados secuencialmente para cada archivo del set de entrenamiento, desarrollo y evaluación. Se realizó la búsqueda de cada fonema en las 6 listas de embeddings y a continuación se concatenaron todos los vectores en uno solo de dimensión $6N$. Si el fonema no existía en la lista de embeddings de algún idioma se le asignó el vector de "missing grams" vector de la misma dimensión N con un valor cercano a cero.

La experimentación partió de la configuración descrita en la Tabla 1 para los parámetros de los embeddings y de la configuración de la Tabla 2 para los parámetros del sistema

LID. En la Tabla 1 se muestra un sistema de bi-gramas que utiliza a Skip-gram (SG) como el modelo utilizado, con una ventana de 8, muestreo negativo de 25, 10 iteraciones y la lista de entrenamiento dividida (el 40% se utilizó en la generación de los embeddings y el 60% como set de entrenamiento para la generación de los i -Vectores). La Tabla 2 muestra el tamaño del vector de cada lista de los embeddings generados en este caso 40, el vector concatenado total fue de 240, el número de gaussianas entrenadas de 128 y la técnica de reducción de dimensionalidad PCA (Principal Components Analysis) de dimensión 100.

"CW/SG" es el modelo utilizado CBOW o Skip-gram, "W" es la ventana de datos del modelo, "NS" el valor para muestreo negativo e "Iter" el número de iteraciones para el algoritmo de Back-Propagation que recalcula los pesos.

Los resultados de la Tabla 4, muestran que un mayor número de gaussianas mejora las métricas por tener un espacio gaussiano más preciso, lo cual era esperado [10]. Por otro lado la Tabla 3 y 5 muestran que PCA no mejora el sistema y aumenta el espacio de almacenamiento requerido por los vectores.

Una vez analizados estos primeros resultados se continuó con otro enfoque para la creación del UBM e i -Vectores, se decidió probar sistemas individuales de cada idioma y fusionarlos, este punto de vista permitió el desarrollo de un nuevo sistema que fue el definitivo. El sistema inicial anterior sin embargo, permitió extraer conclusiones importantes como el número de gaussianas y la pertinencia de PCA

N-gramas	CW/SG	W	NS	Ier	Lista entrenamiento
2	SG	8	25	10	60-40

Tabla 1. Parámetros iniciales para los embeddings

Vector	Vector total	Gauss	Red. Dimensión
40	240	128	PCA 100

Tabla 2. Parámetros iniciales LID

EXP.Nº	Gauss	Vector	Vector total	PCA	Fact	Cavg
1	128	40	240	100	0,5859	0,2665
2	128	40	240	120	0,5830	0,2801
3	128	100	600	100	0,6419	0,2797

Tabla 3. Pruebas de tamaño del vector con PCA

EXP. Nº	Gauss	Fact	Cavg
1a	16	0,7045	0,3037
1b	32	0,6318	0,2860
1c	64	0,5942	0,2731
1	128	0,5859	0,2665
1d	256	0,5644	0,2615
1e	512	0,5288	0,2469

Tabla 4. Pruebas para el número de gaussianas

EXP. Nº	Gauss	Vector	Vector total	PCA	Fact	Cavg
4	32	40	240	120	0,6340	0,2876
5	32	20	120	NO	0,6132	0,2762

Tabla 5. Experimentación sin PCA

EXP. Nº	Gauss	Vector	Vector total	Fact	Cavg
6	32	20	20	0,6879	0,3058

Tabla 6. Experimento inicial para un sistema de un solo idioma

EXP.Nº	Gauss	Vector	Vector total	PCA	Fact	Cavg
7	32	50	50	No	0,7295	0,3281
8	32	10	10	No	0,7727	0,3357

Tabla 7. Pruebas de tamaño del vector sin PCA

EXP.Nº	N-gramas	Vector	Vector total	PCA	Fact	Cavg
9	3	50	50	No	0,7873	0,3445

Tabla 8. Pruebas con trigramas

EXP. Nº	Gauss	Vector	Contex	Fact	Cavg
10	32	20	Yes	0,5946	0,2760

Tabla 9. Pruebas agregando información de contexto

EXP. Nº	Gauss	Vector	Contex	Fact	Cavg
11	32	10	Yes	0,5784	0,2703
12	32	8	Yes	0,6469	0,2959

Tabla 10. Pruebas para el tamaño del vector con contexto

EXP. Nº	Gauss	Vector	PCA	Contex	Fact	Cavg
13	32	10	No	Yes	0,6139	0,2786

Tabla 11. Prueba con la lista de entrenamiento completa

A. SISTEMA FINAL

Para el experimento N°6, se creó un sistema utilizando la plantilla de embeddings obtenida para el idioma Vasco similar al experimento N°5, este sistema es el presentado en la Figura 1. Se utilizó un tamaño de vector de 20 para entrenar directamente el UBM y los i-Vectores. Este nuevo sistema aportó métricas inferiores en un 10%, pero se consideraron mejores desde la perspectiva de que se trató de un solo sistema para un idioma y aún quedaban otros 5 para fusionarlos.

a) Tamaño del vector

Para el experimento N°7 se utilizó un tamaño de vector de 50 y para el experimento N°8 el tamaño fue de 10.

b) N-gramas

Las métricas del experimento N°9 fueron inferiores, el sistema de bi-gramas es mejor que el de tri-gramas en un 7,34% en Fact y en 4,76% en Cavg. Revisamos el mapeo de fonemas en el sistema de tri-gramas y nos percatamos de un número elevado de "missing-grams". Continuamos utilizando al experimento N°6 con bi-gramas como experimento base y dadas las características de los sistemas LID y las propias de los embeddings, introducimos información de contexto para verificar su aporte.

c) Incorporación del contexto

Se introdujo información de contexto en los embeddings, realizando operaciones algebraicas entre cada elemento de los vectores para recalcularlos, utilizando su propia información, la del embedding anterior y del siguiente. La fórmula aplicada para cada vector de la secuencia fonética fue:

$$\text{Vect}'_{\text{act}} = 0,5 * \text{Vect}_{\text{act}} + 0,25 * \text{Vect}_{\text{ant}} + 0,25 * \text{Vect}_{\text{sig}}$$

Donde: $\text{Vect}'_{\text{act}}$ es el vector recalculado, Vect_{act} es el vector actual, Vect_{ant} es el vector anterior y Vect_{sig} es el vector siguiente.

Los resultados obtenidos fueron los presentados en la Tabla 9.

d) Tamaño del vector con contexto

N-gramas	Listas %	CW/SG	W	NS	Iter	Vector	Gauss	Fact	Cavg
2	60-40	SG	8	25	10	10	32	0,5784	0,2703

Tabla 12. Resumen de los parámetros del experimento N°11

EXP. Nº	CW/SG	W	NS	Iter	Fact	Cavg
14	CW	8	25	10	0,6869	0,3204
15	SG	5	25	10	0,6591	0,2955
16	SG	10	25	10	0,5761	0,2623
17	SG	8	10	10	0,6353	0,2833
18	SG	8	35	10	0,5877	0,2808
19	SG	8	25	25	0,6142	0,2801

Tabla 13. Pruebas con diferentes configuraciones en W2V

n gram	Listas %	CW/SG	W	NS	Iter	Vector	Gauss	Fact	Cavg
2	60-40	SG	10	25	10	10	32	0,5761	0,2623

Tabla 14. Parámetros finales con mejores resultados

En la Tabla 10 se muestran los resultados para varios tamaños de vector, los mejores resultados se obtuvieron para vectores de dimensión 10 (experimento N°11). Este tamaño fue el definitivo para las siguientes experimentaciones, y concluimos con la determinación de los parámetros del sistema LID.

e) Tamaño del set de entrenamiento dedicado a los embeddings

Se repitió el experimento N°11 variando el set de entrenamiento, utilizando las listas completas tanto para entrenar el sistema como para generar los embeddings, es decir, sin bloques superior e inferior.

Los resultados fueron ligeramente inferiores, además que el tiempo de procesamiento incrementó considerablemente, esto se debe a que el sistema se sobreentrena. Por tanto, se eligió trabajar a partir de este punto con listas de entrenamiento divididas 60%-40%. Las siguientes pruebas se realizaron para establecer la mejor configuración de los modelos CBOW y Skip-gram.

f) Parámetros CBOW/Skip-gram

La Tabla 12 muestra un resumen completo de los parámetros óptimos obtenidos hasta el momento, corresponden al experimento N°11.

Realizamos los cambios de un parámetro a la vez, manteniendo invariables los demás. Se muestran en la Tabla 13 los valores de Fact y Cavg para esta experimentación (se resalta el parámetro modificado).

B. EXTRACCIÓN DE PARÁMETROS ÓPTIMOS

De las pruebas del apartado anterior se obtuvieron los parámetros que proporcionan los mejores resultados. Se observó un cambio positivo únicamente en el tamaño de la ventana, los demás parámetros no aportaron y, por tal motivo, los parámetros finales seleccionados fueron:

Estas métricas se han obtenido para el HMM (Húngaro), un idioma (Vasco) y para 32 gaussianas.

El sistema final propuesto consistió en realizar la fusión de los seis sistemas correspondientes a cada uno de los seis idiomas

y con 512 gaussianas con la mejor configuración que se ha mostrado en la Tabla 14.

EXP. N°	Idioma	Fact	Cavg
20	Vaco	0,5125	0,2438
21	Catalán	0,4969	0,2440
22	Inglés	0,5077	0,2451
23	Gallego	0,5195	0,2514
24	Portugués	0,5416	0,2623
25	Español	0,5257	0,2544
26	Fusión	0,3754	0,1870

Tabla 15. Métricas para los 6 idiomas y su fusión

EXP. N°	SISTEMA 1	SISTEMA 2	Fact	Cavg	Improv. Fact	Improv. Cavg
27	None	Embedding Vasco	0,5125	0,2438	NA	NA
28	PPRLM 3-gram Hung	None	0,3284	0,1716	NA	NA
29	PPRLM 3-gram Hung	Embedding Vasco	0,2917	0,1553	11,2%	9,5%
30	PPRLM 3-gram Hung	6 Embeddings	0,2764	0,1493	15,8%	12,9%

Tabla 16. Fusión de los sistemas LID Embeddings-PPRLM

Los datos obtenidos para cada sistema de un idioma (experimentos N°20 al N°25) reflejan métricas que están dentro de un mismo rango, los valores de Fact y Cavg son relativamente elevados en comparación con otros sistemas LID.

El experimento N°26 es la fusión de los 6 sistemas de cada idioma. Claramente se observaron mejoras en las métricas finales producto de la comparación entre la fusión de los 6 sistemas con el mejor de ellos, en este caso Vasco.

IV. RESULTADOS DE LA FUSIÓN

La experimentación final realizada, mostrada en la Tabla 16, fue la fusión de estos sistemas de embeddings con otros de diferente naturaleza y verificar su aportación.

La fusión se ejecutó sobre un sistema PPRLM basado en un modelo de 3-gramas con suavizado Witten Bell existente. El suavizado permite evitar las probabilidades cero de los n-gramas no vistos en el texto, redistribuyendo estas probabilidades y compensando a los n-gramas menos frecuentes o no vistos. A diferencia de otras técnicas de suavizado, utiliza como información adicional el contexto.

El experimento N°27 replica los resultados obtenidos en el experimento N°20 sin fusiones con otros sistemas, en el experimento N°28 están únicamente los resultados para el sistema PPRLM para 3-gramas que ha presentado los mejores resultados.

Para el experimento N°29 se han fusionado el sistema PPRLM y el sistema de embeddings con mejores métricas, en este caso Vasco. Presentaron una mejora respecto al sistema PPRLM de 11,2% en Fact y 9,5% en Cavg.

El último experimento N°30 muestra los resultados obtenidos para la fusión del sistema PPRLM con los 6 sistemas de embeddings, la mejora fue de 15,8% en Fact y 12,9% en Cavg.

Estos resultados demuestran la aportación positiva de los sistemas de embeddings cuando se combinan con otros sistemas.

V. CONCLUSIONES

La metodología secuencial de los experimentos permitió cubrir un amplio rango de las posibilidades sobre la utilización de la representación vectorial a nivel fonético.

Los modelos propuestos Continuous Bag of Words y Skip-gram han sido desarrollados inicialmente para la representación de unidades fonéticas complejas como palabras, sin embargo se ha propuesto emplear unidades más pequeñas como n-gramas que, de acuerdo a las pruebas realizadas con la base de datos disponible, presentó mejores resultados con unidades más pequeñas (2 gramas) porque el vocabulario generado incluye la mayoría de combinaciones posibles. Los poli-fonemas de mayor dimensión fueron más difíciles de abarcar durante el entrenamiento de los embeddings y produjeron más “missing-grams” o fonemas no encontrados, que en el modelo se reemplazaron con un vector de valor cero. Esta observación se tradujo en el aumento de la tasa de error. Una alternativa para realizar pruebas con n-gramas mayores podría ser la utilización de una base de datos mayor para la generación de los embeddings.

Los datos del vector contienen información muy dispersa para las unidades fonéticas empleadas, por lo que la tendencia de reducir el tamaño del vector generó mejores resultados y determinó el mínimo permitido por el reconocedor (dim. 10).

La configuración del modelo de un solo idioma, se la aplicó a los otros 5 para disponer de seis sistemas que al final fueron fusionados para obtener las métricas finales.

El sistema propuesto por sí solo está por debajo de otros sistemas LID, pero su fusión con otras técnicas puede contribuir significativamente como se ha demostrado.

REFERENCIAS

- [1] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu. “Language Identification: A tutorial”. *IEEE Circuits and Systems Magazine*, pages 82-108. May 2011.
- [2] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds. “Acoustic, phonetic, and discriminative approaches to automatic language identification”. *Interspeech*, 2003.
- [3] C. Salamea, L.F. D'Haro, R. de Córdoba, M. A., Caraballo “Incorporación de n-gramas discriminativos para mejorar un reconocedor de idioma fonotáctico basado en i-vectores”, *Procesamiento del Lenguaje Natural, Revista n° 51*, páginas 145-152, 2013.
- [4] M. A. Zissman et al. “Comparison of four approaches to automatic language identification of telephone speech”. *IEEE Transactions on Speech and Audio Processing*, pages 31 - 44, 1996.
- [5] L. J. Rodríguez-Fuentes, N. Brummer, M. Penagarikano, A. Varona, G. Bordel, and M. Díez. “The Albayzin 2012 language recognition evaluation”. In *Interspeech*, pages 1497 - 1501, 2013.
- [6] S. Lai, K. Liu, L. Xu and J. Zhao. “How to Generate a Good Word Embedding”, National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, China, July 2015.
- [7] T. Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. “Distributed Representations of Words and Phrases and their Compositionality”. In *Proceedings of NIPS*, 2013.
- [8] M. Díez, A. Varona, M. Peñagarikano, L. J. Rodríguez-Fuentes, and G. Bordel. “On the use of phone log-likelihood ratios as features in spoken language recognition”. In *SlT*, pages 274-279, 2012.
- [9] P. Schwarz, “Phoneme Recognition based on Long Temporal Context”, PhD Thesis. Brno University of Technology, 2009.
- [10] D. A. Reynolds. “A Gaussian mixture modeling approach to text independent Speaker identification”. Ph.D. thesis, Georgia Inst. of Technol., 1992.



Francisco Charro, Ingeniero en Electrónica y Telecomunicaciones por la Escuela Politécnica Nacional (EPN) 2006, Diplomado Superior en Plataformas Operativas e Internetworking por la Escuela Politécnica Nacional 2008, Máster Universitario en Sistemas Electrónicos por la Universidad Politécnica de Madrid (UPM) 2015. Docente a tiempo completo en la Escuela Politécnica Nacional y a tiempo parcial en la Universidad de las Américas desde 2016.



Marco Herrera, Ingeniero en Electrónica y de Control por la Escuela Politécnica Nacional (EPN) 2009 y Master Universitario en Automática y Robótica por la Universidad Politécnica de Madrid (UPM), España, en 2014. De 2010 a 2012, docente en la Universidad Internacional del Ecuador, Quito, Ecuador. Desde 2014 es docente en el Departamento de Automatización y Control Industrial de la Escuela Politécnica Nacional. Su interés de investigación incluye control óptimo, control de modo deslizante y modelado borroso.



Nataly Pozo, Ingeniera en Electrónica y Control en la Escuela Politécnica Nacional. A partir de entonces se trabajó como Técnico Docente Politécnico (DACI - EPN). En el año 2015- 2016 estudió en la Universidad de Newcastle – Reino Unido, en donde obtuvo el título de M.Sc. Electrical Power. Actualmente se desempeña en el cargo de Profesora a Tiempo Completo en el Departamento de Automatización y Control Industrial de la EPN. Sus áreas de interés son: Electrónica de Potencia, Convertidores Estáticos, Energía Renovable, Redes Eléctricas Inteligentes, Diseño y Control de Máquinas.



Andrés Rosales, Profesor Principal del Departamento de Automatización y Control Industrial. Ingeniero en Electrónica y Control y Doctor en Ingeniería en Sistemas de Control (Universidad Nacional de San Juan – UNSJ, Argentina, 2009). Investigador invitado (Universidad de Hannover, Alemania, 2007-2008). Actualmente, Director de Investigación y Proyección Social de la EPN. Fue Coordinador General Técnico del CEAACES (2014-2015), y Editor de la Revista Politécnica de la EPN (2012-2014).