

Transcripción fonética de acrónimos en castellano utilizando el algoritmo C4.5

Carlos Monzo, Francesc Alías, Jose Antonio Morán, Xavier Gonzalvo

Dpto. Comunicaciones y Teoría de la Señal
Enginyeria i Arquitectura La Salle – Universidad Ramon Llull
C/ Quatre Camins 2, 08022 Barcelona (España)
{cmonzo, falias, moran, gonzalvo}@salle.url.edu

Resumen: Este trabajo presenta un transcriptor automático de acrónimos con el objetivo de incrementar la calidad de la síntesis generada en un conversor de texto en habla, ante la presencia de acrónimos en el texto. La transcripción de los acrónimos se realiza usando un árbol de decisión (algoritmo C4.5) sobre los datos de entrenamiento. El trabajo presenta los resultados obtenidos para diferentes configuraciones del algoritmo, comparando su funcionamiento respecto a otros sistemas de aprendizaje.

Palabras clave: acrónimo, transcripción automática, PLN, conversión texto-habla, C4.5, aprendizaje artificial, WEKA, Soundex, Greedy

Abstract: This work presents an automatic acronyms transcription system in order to increase the synthetic speech quality of text-to-speech systems, in the presence of acronyms in the input text. The acronyms transcription is conducted by using a decision tree (C4.5 algorithm). The work presents the results obtained for different algorithm configurations, validating its performance with respect to other learning systems.

Keywords: acronym, automatic transcription, NLP, text-to-speech, C4.5, artificial learning, WEKA, Soundex, Greedy

1 Introducción

Los sistemas de conversión de texto en habla (CTH) generan un mensaje oral a partir de un texto de entrada. El sistema consta de dos bloques, un bloque de procesamiento del lenguaje natural (PLN) y otro de procesamiento digital de la señal (PDS). El primero es el encargado de generar la transcripción fonética y la prosodia del texto, mientras que el segundo toma esta información para generar el mensaje oral correspondiente.

La transcripción fonética se encarga de la conversión de un texto (grafemas) a sus fonemas¹ correspondientes. Esta transcripción

se realiza habitualmente mediante la aplicación de reglas (p. ej. castellano) o el uso de diccionarios (p. ej. inglés).

Uno de los elementos clave del transcriptor fonético es el proceso que realiza la normalización del texto. Este proceso es el encargado de tratar los números, abreviaturas y acrónimos para obtener la transcripción fonética correspondiente a sus equivalentes escritos o extendidos (p. ej. 12 → doce o Sr. → señor). La normalización se puede realizar a partir de diccionarios de excepciones, donde se guarda la conversión de grafema a fonema, con el problema de que por la poca flexibilidad de éste, para cualquier cambio en el corpus se debe realizar una comprobación manual. Por contra, automatizando del proceso de normalización, el tratamiento de las excepciones se realizaría de modo transparente al resto del CTH.

¹ Estos símbolos se representan en este trabajo mediante símbolos SAMPA (Wells, 1999)

Por acrónimo se entiende aquella palabra que se forma a partir de las iniciales o partes de otras palabras (p.ej. PSOE o FBI). Este trabajo se centra en la transcripción fonética automática de los acrónimos del texto, para evitar que una incorrecta transcripción haga disminuir la calidad de la síntesis del habla generada.

En este trabajo se presenta el diseño de un transcriptor automático de acrónimos para que el CTH disponga de la correcta pronunciación de los mismos sin la necesidad de disponer previamente de los acrónimos que se emplearán, ya que una vez el transcriptor ha sido entrenado pasa a ser un transcriptor de acrónimos genérico.

En el apartado 2 se describe la base metodológica del método presentado. El apartado 3 detallada la solución propuesta para la transcripción automática de acrónimos, validándose mediante un conjunto de experimentos en el apartado 4. Finalmente, en el apartado 5 se discuten las conclusiones alcanzadas y futuras líneas de trabajo.

2 Base metodológica

En este apartado se presentan los trabajos previos que han servido de base para afrontar el problema, en términos de: *i)* la información usada como entrada al sistema; *ii)* la técnica más adecuada a aplicar a partir del problema planteado y *iii)* la posible codificación de la información de entrada para aumentar la eficiencia del sistema.

2.1 Información de entrada

En el proyecto NETtalk (Sejnowski y Rosenberg, 1987) se utiliza un sistema basado en inteligencia artificial para transcribir fonéticamente palabras en inglés. El aprendizaje se efectúa haciendo uso de una red neuronal (perceptrón multicapa con entrenamiento *back-propagation*) (ver Figura 1).

Del análisis de este trabajo se deduce que la opción más viable, para abordar el problema de qué información se debe pasar al transcriptor automático de acrónimos, son los diferentes grafemas que los forman y la posible relación con el resto.

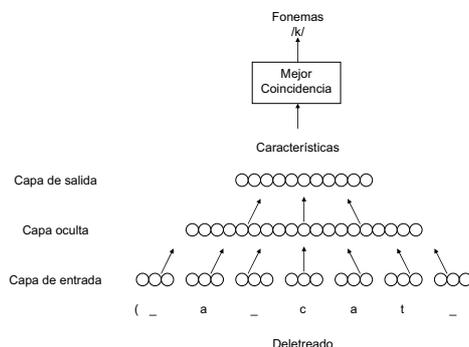


Figura 1: Perceptrón multicapa en NETtalk (Sejnowski y Rosenberg, 1987)

2.2 Técnica a aplicar

La obtención de un sistema de transcripción automática de acrónimos se divide en dos partes, la primera hace una desambiguación de los textos y la segunda obtiene los fonemas más adecuados para ese acrónimo. El presente trabajo se centra en esta segunda fase de forma que necesita de un preprocesamiento que indique si la palabra de interés se trata o no de un acrónimo. Fundamentalmente, existen dos opciones para el etiquetado: *i)* utilizar etiquetas XML o equivalentes cuando se conocen a priori los textos a sintetizar (Alías, 2005); *ii)* utilizar un sistema automático de desambiguación de textos (Mikheev, 2002).

En este trabajo se utilizan árboles de decisión para asociar a un grafema de entrada una cierta pronunciación (“clasificación”) siguiendo el trabajo de Mikheev (2002), en el que clasifica (desambigua) los textos mediante esta técnica de aprendizaje (ver Figura 2).

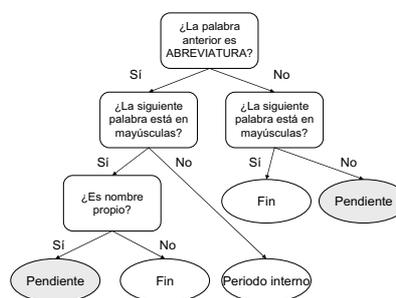


Figura 2: Uso de árboles de decisión desambiguar abreviaturas (Mikheev, 2002)

2.3 Codificación de la información

Los resultados obtenidos en sistemas de recuperación de información, al utilizar la codificación sobre los textos de interés, usando Soundex (NARA, 1995) hacen plantear que la misma idea puede ser aplicada al algoritmo de aprendizaje usado en la transcripción automática de acrónimos, con el objetivo de reducir el volumen de información con el que debe trabajar. La idea es codificar los grafemas de entrada consiguiendo que la información que tiene características similares pueda ser agrupada. En Soundex la agrupación se realiza mediante seis dígitos asociados a seis clases sonoras distintas (ver Tabla 1).

En este trabajo se estudiará si la compactación de la información siguiendo la relación grafema/sonoridad permite el aumento de su eficiencia. Para ello se deberá de adaptar la codificación Soundex definida para el inglés al conjunto de grafemas/sonoridad del castellano (ver apartado 3.4).

Dígito	Grafema representado
1	B, F, P, V
2	C, G, J, K, Q, S, X, Z
3	D, T
4	L
5	M, N
6	R

Tabla 1: Tabla de codificación Soundex (NARA, 1995)

3 Descripción del sistema

A continuación se pasa a detallar el transcriptor automático de acrónimos. Se muestran los diferentes planteamientos realizados para abordar el problema, el algoritmo utilizado y las soluciones alcanzadas.

3.1 Planteamiento del problema

El objetivo que se persigue es el de disponer de un sistema automático de transcripción de acrónimos (que realizará la normalización de los textos dentro del bloque de PLN) en lugar del habitual diccionario de excepciones.

Estos acrónimos provienen de un bloque anterior del sistema que será el responsable de clasificar, de entre todos los textos de entrada, qué palabra se debe tratar como acrónimo y cuál no. Este bloque puede ser tanto un

clasificador automático como directamente el texto marcado utilizando etiquetas del estilo XML. En este trabajo se parte de la hipótesis que alguno de los métodos de desambiguación comentados ha sido utilizado previamente.

El caso de los acrónimos es problemático debido a que no ocurre como en otros casos, como por ejemplo números, en los que la normalización es más sencilla por conocerse de antemano cómo se debe actuar para cada nuevo caso que se presente. El problema de los acrónimos radica en su lectura, debido a que no siguen unas reglas predefinidas por ser en cada caso la combinación de grafemas quien la fijará. Las diversas opciones para ser leídos son las siguientes:

1. Normal. Como una palabra más de la lengua (p. ej. ONU)
2. Deletreando la palabra (p. ej. FBI)
3. Lectura híbrida. Lectura normal más deletreo (p. ej. PSOE)

A partir de técnicas de aprendizaje artificial se extraerá el conocimiento necesario para generalizar el obtenido de los ejemplos de entrenamiento. Entre las técnicas que pueden ser utilizadas, como se ha visto en el apartado 2, se ha optado por el uso árboles de decisión.

3.2 Algoritmo de aprendizaje utilizado

Entre las distintas aproximaciones para el modelado de los datos mediante árboles de decisión se ha seleccionado usar el algoritmo C4.5 (Quinlan, 1993), principalmente por la simplicidad que presenta su funcionamiento frente a otros (una vez demostrada su utilidad se podrá plantear la posibilidad de complicar los diseños). Para su aplicación se ha utilizado la herramienta informática WEKA (Witten y Frank, 2005), donde se pueden encontrar multitud de algoritmos de aprendizaje artificial.

El algoritmo C4.5 (J4.8 en WEKA), se trata de un árbol de decisión que acepta tanto datos numéricos como nominales. Permite realizar poda de sus ramas consiguiendo así variar el grado de generalización de la solución obtenida, correspondiéndose una mayor poda con una mayor generalización.

Las distintas configuraciones se muestran en el apartado 4. Las diferentes propuestas han sido diseñadas con el objetivo de aumentar la eficiencia del algoritmo y maximizar los porcentajes de clasificación.

3.3 Datos de entrenamiento

3.3.1 Preparación de los datos

Con el objetivo de maximizar la presencia de los diferentes grafemas en la etapa de aprendizaje se realiza la selección de los datos que serán utilizados en el proceso de entrenamiento.

Como primer paso se recopilan los acrónimos que puedan ser de utilidad, procurando que formen parte de distintos dominios para intentar así dar mayor variabilidad a los datos de entrenamiento. Los dominios considerados de mayor interés, por el volumen de acrónimos que se suelen encontrar son: “Tecnología” (informática e Internet) y “Periodismo” (política y economía), añadiéndose a ellos una serie de “Varios”, que dan cobertura a otros ámbitos como son educación, medicina... La Tabla 2 presenta los diferentes dominios y su cobertura (porcentaje de pertenencia a cada uno de ellos).

Dominio	Número de representantes	Cobertura
Tecnología	1862	54 %
Periodismo	1432	41 %
Varios	174	5 %
Total	3468	100 %

Tabla 2: Dominios disponibles y su cobertura

El siguiente paso consiste en seleccionar el subconjunto de acrónimos para entrenar al sistema. Los datos de entrenamiento deben disponer de las parejas acrónimo-transcripción fonética para que el algoritmo C4.5 sea capaz de aprender y generalizar sus correspondencias. Esta información será generada por un experto, de forma que se indique para cada grafema la transcripción (salida) asociada.

A priori se desconoce la correspondencia entre grafema y su transcripción fonética ($P \rightarrow /pE/$ o $/p/$), por tanto el procedimiento que se sigue es el de asegurar un número mínimo de apariciones de los grafemas dentro de los acrónimos, de igual forma que también se tiene en cuenta su tamaño. La idea de considerar el tamaño (número de grafemas) se basa en que no será lo mismo uno considerado largo (p. ej. 6 grafemas), que tendrá más posibilidades de que ciertas partes se lean de forma normal o híbrida, que uno corto (p. ej. 3 grafemas) donde lo más habitual será que deba deletrearse.

3.3.2 Elección de los datos

Para llevar a cabo la elección de los datos se utilizará un algoritmo Greedy (François y Boëffard, 2002) sobre el conjunto de datos disponible. Este algoritmo selecciona los acrónimos que cumplen con una serie de requisitos de entrada, siendo éstos el número de apariciones por grafema y su tamaño (criterio de corto y largo).

Por lo que hace referencia a los grafemas, se utilizan los 27 del castellano. Al algoritmo se le pasará esta información, es decir el listado con los grafemas de interés, juntamente con el número de apariciones mínimo que se desea que haya de cada uno de ellos (ajustado empíricamente a 10 apariciones por grafema en acrónimos cortos y 10 más en largos disponiendo así de un volumen de información que permita trabajar con ella). El algoritmo Greedy consigue minimizar el número de acrónimos seleccionados si previamente se ordena la lista de grafemas en orden ascendente según la distribución de apariciones obtenidas a partir de la información de entrada.

Una vez realizada esta selección, se pasa a elegir un solo ejemplo en el caso que haya redundancias (eliminando el resto buscando cada acrónimo entre los que han sido seleccionados y comparándolo con el resto) debidas a haber usado dos criterios (cortos y largos), obteniéndose finalmente 281 acrónimos con un total de 1099 grafemas con los que se realizará el entrenamiento y testeo. La distribución de grafemas obtenida es la que se observa en la Figura 3.

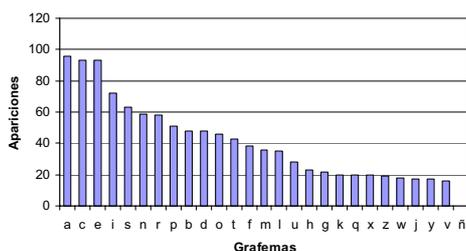


Figura 3: Aparición de grafemas en el entrenamiento

En el caso concreto de la “ñ” no se ha encontrado ningún acrónimo que contenga ese grafema.

3.4 Procesamiento de la información

Decididos los datos de entrenamiento se pasa a realizar la transcripción para cada grafema, obteniéndose 48 transcripciones diferentes (clases de salida). Estas transcripciones corresponden a cómo se pronunciará en cada caso el grafema de interés (p. ej. FBI → /Efe/ /BE/ /I/). Para estas transcripciones se ha utilizado notación SAMPA modificada, donde la tonicidad de las vocales se indica con su versión en mayúsculas.

La idea es que cada grafema que forma el acrónimo disponga de una representación en el espacio de transcripciones, por tanto el número de grafemas coincidirá con el de símbolos representando la transcripción (facilitando el proceso de pasar la información al algoritmo). Si por cualquier razón alguno de sus grafemas no tiene transcripción le será asociado el símbolo “/” (p. ej. HI-FI → /-/ /I/ /-/ /f/ /i/). Cada transcripción se identifica por la colocación de una “/” delante y detrás, y un espacio en blanco que las separa (p. ej. PSOE → /pE/ /s/ /O/ /e/). A aquellos grafemas del acrónimo que sean “/” o “-” se les asocia “/”.

Como se verá en el apartado 4, se han realizado dos versiones de transcripción y dos tipos de pruebas dentro de éstas. La primera es aquella donde los grafemas, como por ejemplo “b” o “d”, a principio de palabra tienen una transcripción distinta a la que tienen en cualquier otra posición dentro de la frase (reglas del castellano. P.ej. BCC → /bE/ /TE/ /TE/ o /BE/ /TE/ /TE/ respectivamente). Este hecho no debería tenerse en cuenta debido a que en la gran mayoría de casos los acrónimos se encuentran en cualquier posición dentro del texto. La otra consideración a realizar es que en las transcripciones de vocales se incluye la información de tonicidad. Esto complica el proceso de aprendizaje por aumentar la dimensionalidad de los datos a ser tratados de forma que se harán pruebas sin considerar dicha información.

Los datos se pasan al algoritmo de aprendizaje mediante un ventaneo del acrónimo, de forma que el de interés queda posicionado en el centro de la ventana (tamaño impar). El tamaño será un parámetro a estudiar para ver la dependencia con los resultados obtenidos. Mediante “#” se indicará que el grafema de interés está en un extremo del acrónimo (proceso ejemplificado en la Tabla 3).

Abrev.	Graf. interés	Graf. anterior	Graf. posterior	Transcr.
zcs	z	#	c	/TEta/
	c	z	s	/TE/
	s	c	#	/Ese/

Tabla 3: Aplicación de una ventana de tamaño igual a 3 sobre los datos de entrenamiento

Una excesiva variabilidad de los datos de entrada al árbol de decisión, al trabajar con todas las combinaciones que pueden darse en los acrónimos de entrenamiento, puede provocar que los resultados de clasificación no sean satisfactorios (ver apartado 4). Se plantea disminuir la dimensionalidad del problema aplicando una codificación sobre los grafemas anteriores y posteriores al de interés. Con este fin se utiliza la codificación Soundex, presentada en el apartado 2, adaptándola al castellano y denominándola Soundesp. En la Tabla 4 se puede ver la versión inicial, más próxima a Soundex, y en la Tabla 5 la versión final, donde la agrupación se realiza mediante un ajuste más fino de los sonidos con comportamiento similar.

Código	Grafema
UNO	B F P V
DOS	C G J K Q S X Z
TRES	D T
CUAT	L
CINC	M N Ñ
SEIS	R
SIET	H
OCHO	A E I O U W Y
NUEV	/ - #

Tabla 4: Versión inicial de Soundesp

Código	Grafema
UNO	B P V D T K
DOS	C J S Z L R
TRES	X Q
CUAT	F G
CINC	M N Ñ
SEIS	H
SIET	A E I O U
OCHO	W
NUEV	Y
DIEZ	/ - #

Tabla 5: Versión final de Soundesp

Para terminar, en la Tabla 6 se muestra un ejemplo de la aplicación de Soundesp (versión final) sobre un acrónimo usando una ventana de tamaño igual a 3.

Acron.	Graf. interés	Graf. anterior	Graf. posterior	Transcr.
zcs	z	DIEZ	DOS	/TEta/
	c	DOS	DOS	/TE/
	s	DOS	DIEZ	/Ese/

Tabla 6: Codificación Soundesp (versión final) de “zcs”

4 Análisis de resultados

A lo largo de este apartado se presentan el conjunto de pruebas realizadas, descritas a continuación:

- Barrido del tamaño de la ventana: 3, 5 y 7 grafemas
- Variación en la representación de los datos: sin utilizar ningún tipo de codificación sobre los grafemas anterior y posterior al de interés y utilizando la versión inicial y final de Soundesp
- Uso de la información de tonicidad de las vocales
- Una vez decidida la mejor configuración utilizando el algoritmo C4.5 con poda del árbol y *10 fold Cross-Validation* para el testeo, se pasarán a validar los resultados obtenidos sin aplicar poda y aplicando IBk (Aha, Kibler y Albert, 1991) y NaiveBayes (John y Langley, 1995)

4.1 Porcentajes de clasificación

La medida empleada, para decidir la mejor configuración, es la de tanto por ciento de instancias correctamente clasificadas (es decir las transcripciones que el sistema ha realizado correctamente).

Otros parámetros serán tenidos en cuenta a la hora de ser aplicado el árbol de decisión, como es el caso de F1 (apartado 4.2) que da una visión sobre la precisión y grado de cobertura y estos dos de manera particular. Las ecuaciones que se muestran en (1), (2) y (3) (Sebastiani, 2002) expresan el cálculo tal y como se realiza internamente en WEKA.

$$F1 = \frac{2 \cdot \text{precisión} \cdot \text{cobertura}}{\text{precisión} + \text{cobertura}} \quad (1)$$

$$\text{precisión} = \frac{VP}{VP + FP} \quad (2)$$

$$\text{cobertura} = \frac{VP}{VP + FN} \quad (3)$$

Donde VP = ‘Verdadero Positivo’ (clasificación correcta), FP = ‘Falso Positivo’ (clasificación incorrecta debido a asociar la salida a la clase cuando esto no es cierto) y por último FN = ‘Falso Negativo’ (clasificación incorrecta debido a asociar la salida a una clase diferente no siendo cierto). El objetivo es conseguir una alta precisión y cobertura en la clasificación, es decir que F1 sea máxima.

	Trans1	Trans2
Grafemas 3	78,82	77,92
Soundesp ini - 3	84,38	83,21
Soundesp final - 3	84,29	83,84
Grafemas 3 acentos	67,15	66,25
Soundesp ini - 3 acentos	72,08	72,44
Soundesp final - 3 acentos	72,44	72,53
Grafemas 5	78,28	77,83
Soundesp ini - 5	82,5	82,56
Soundesp final - 5	83,75	83,12
Grafemas 5 acentos	67,15	66,16
Soundesp ini - 5 acentos	70,65	71,10
Soundesp final - 5 acentos	71,81	72,35
Grafemas 7	78,64	78,01
Soundesp ini - 7	81,68	81,43
Soundesp final - 7	82,41	82,23
Grafemas 7 acentos	66,97	65,98
Soundesp ini - 7 acentos	70,65	71,63
Soundesp final - 7 acentos	72,80	73,25

Tabla 7: Porcentaje de instancias correctamente clasificadas para diferentes transcripciones

En la Tabla 7 se muestran los valores porcentuales de instancias correctamente clasificadas. La primera columna hace referencia a: *i*) la configuración utilizada en cuanto a tamaño de la ventana (3, 5 ó 7); *ii*) la codificación aplicada en el caso de haber sido utilizada (inicial como “ini” o final como “final”); *iii*) si se tiene en cuenta la tonicidad de las vocales del acrónimo (“acentos”). Por ejemplo, “Soundesp ini - 7 acentos” corresponde al caso de haber usado la

codificación Soundesp inicial, una ventana de 7 grafemas y haber tenido en cuenta la tonicidad de las vocales. La segunda y tercera columna (“Trans1” y “Trans2”) indican el planteamiento hecho al transcribir, siendo “Trans1” aquella donde se considera que el acrónimo está a principio de frase, mientras que en “Trans2” se considera que puede estar en cualquier posición dentro del texto.

Como se observa en la Tabla 7 no existen diferencias importantes por el hecho de usar cualquiera de las dos opciones, por tanto será “Trans2” la configuración utilizada para el resto de pruebas.

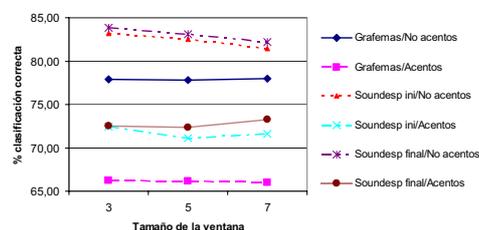


Figura 4: Porcentaje de instancias correctamente clasificadas

Usando “Trans2” se genera la Figura 4 donde se muestra la mejor configuración para tratar la información de entrada al algoritmo. Se observa que entre las diferentes codificaciones “Soundesp final” proporciona una mejora sobre el porcentaje de clasificación correcta (un aumento de aproximadamente el 6% respecto no usar codificación). Por otro lado, de este análisis se puede concluir que el tamaño de las ventanas no es un parámetro determinante por no existir grandes diferencias en los resultados obtenidos.

4.2 F1

Teniendo en cuenta que la mayor mejora se consigue para una ventana de 3 grafemas se pasan a estudiar los resultados conseguidos en términos de F1 sin considerar la información de la tonicidad (ver Figura 5).

La Figura 5 muestra una gráfica de valores discretos donde los puntos han sido unidos para facilitar su visualización. De los pasos por cero solamente comentar que son aquellas clases que el algoritmo no puede clasificar por no haber podido “aprender” qué debía hacer, especialmente por el hecho que en los datos de

entrenamiento no aparecen casos que sí lo hacen en los de testeo (recordemos que se utiliza *10 fold Cross-Validation*) por ser poco representativos. Se puede observar como los mejores resultados, siendo estos los más próximos a 1, son para el caso de Soundesp, siendo éstos similares para las dos configuraciones analizadas y generalmente mejores que para el caso de no usar ninguna codificación.

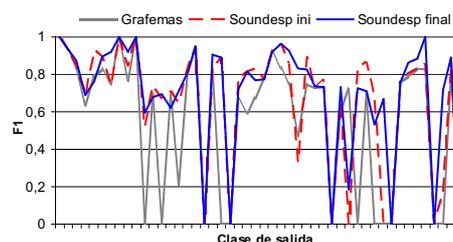


Figura 5: F1 para las 48 clases de salida. En el eje X se indican las clases de salida posibles (transcripciones de los grafemas)

4.3 Validación de la propuesta

Una vez presentados los resultados aplicando el algoritmo C4.5, se muestran los resultados obtenidos en base a realizar la comparativa utilizando una ventana de tamaño igual a 3 grafemas y las configuraciones de codificación de los datos de entrada “Grafemas”, “Soundesp ini” y “Soundesp final” sobre un barrido de algoritmos: C4.5 con y sin poda del árbol, IBk y NaiveBayes; utilizando para ello la medida de testeo *10 fold Cross-Validation* (ver Figura 6). Los valores de configuración para cada uno de los algoritmos es la que tiene WEKA por defecto.

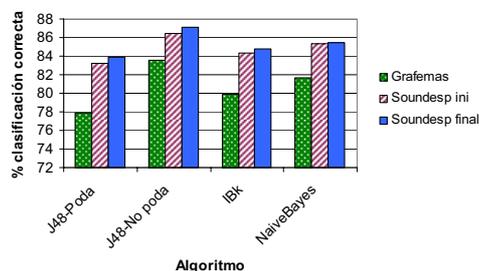


Figura 6: Porcentaje de clasificación correcta para diferentes algoritmos de aprendizaje

De la Figura 6 se desprende que el uso de Soundesp (versión inicial o final) aumenta el rendimiento del sistema para cualquiera de los métodos de aprendizaje utilizado. Asimismo se observa que el hecho de no aplicar poda en C4.5 mejora los resultados.

Así pues, para terminar, la configuración que ha presentado mejores resultados a lo largo de las pruebas es:

- Tamaño de ventana = 3
- Transcripción: sin acentos
- Codificación: Soundesp final

5 Conclusiones y líneas de futuro

En este trabajo se ha presentado un transcriptor fonético orientado a la transcripción automática de acrónimos en castellano, con el objetivo de aumentar la calidad del habla generada por un CTH a partir de la correcta lectura de los mismos.

Uno de los elementos claves del sistema es que se evita la necesidad de disponer de un diccionario orientado a cubrir las limitaciones del transcriptor utilizado, teniendo en cuenta que debido a la poca flexibilidad que presentan frente a nuevos casos su uso es poco eficiente.

El algoritmo C4.5 (J4.8 en WEKA) es capaz de transcribir correctamente hasta un 85% de los casos de entrada, viendo así la utilidad que tiene añadir al bloque de PLN un módulo como el presentado. Visto el correcto funcionamiento se podría profundizar en el estudio de otros algoritmos de aprendizaje, tal y como se ha introducido en el apartado 4, para intentar mejorar los porcentajes de clasificación.

Se ha observado que para este tipo de aplicaciones, el hecho de codificar los datos de entrada, reduciendo así la dimensionalidad del problema, permite incrementar el porcentaje de acrónimos correctamente transcritos. De la misma manera que se ha hecho para castellano, puede pensarse en adaptar la codificación a otras lenguas.

Como línea de futuro, sería interesante trabajar con esas clases de salida con una baja representación, para tratar así de obtener un incremento en las prestaciones.

Agradecimientos

Este trabajo ha sido realizado con el apoyo del proyecto SALERO (IST-FP6-027122) de la Comisión Europea.

Bibliografía

- Aha, D., D. Kibler y M. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, volumen: 6, páginas 37-66.
- Alías F., I. Iriundo, Ll. Formiga, X. Gonzalvo, C. Monzo y X. Sevillano. 2005. High quality Spanish restricted-domain TTS oriented to a weather forecast application. *Interspeech*. Lisboa (Portugal).
- François, H. y O. Boëffard, 2002. The greedy algorithm and its application to the construction of a continuous speech database. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation, LREC*, volumen 5.
- John, G.H. y P. Langley. 1995. Estimating Continuous Distributions in Bayesian Classifiers. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, San Mateo, páginas 338-345.
- Mikheev, A. 2002. Periods, Capitalized Words, etc. *Association for Computational Linguistics*, volumen 28, número 3, páginas 289-318, September.
- NARA. 1995. *Using the Census Soundex*. U.S. National Archives and Records Administration. Washington, DC.
- Quinlan, R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.
- Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, volumen 34, número 1, páginas 1-47, March.
- Sejnowski T.J. y C.R. Rosenberg. 1987. Parallel networks that learn to pronounce English text. *Journal of Complex Systems*, 1(1): 145-168, February.
- Wells, J.C. 1997. *SAMPA computer readable phonetic alphabet*. In Gibbon, D., Moore, R. and Winski, R. (eds.). *Handbook of Standards and Resources for Spoken Language Systems*. Berlin and New York: Mouton de Gruyter. Part IV, section B.
- Witten, I.H. y E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco.