Restauración automática de acentos ortográficos en adverbios interrogativos

Santiago Iturriaga

Facultad de Ingeniería,
Universidad de la República,
Montevideo, Uruguay
siturria@fing.edu.uy

Resumen La omisión de los acentos ortográficos es uno de los errores tipográficos más frecuentes en el idioma español. La restauración automática de acentos ortográficos consiste en la inserción de acentos ortográficos omitidos en los lugares que son necesarios. Los adverbios interrogativos son un caso especialmente dificultoso del problema de restauración automática de acentos ortográficos ya que en muchas ocasiones no existen marcas claras que indiquen la presencia del adverbio interrogativo. Este trabajo presenta dos técnicas de aprendizaje automático aplicadas a la restauración automática de acentos ortográficos para el caso específico de los adverbios interrogativos. La primer técnica aplicada al problema es conditional random fields (CRF), un modelo introducido por Lafferty, McCallum, y Pereira[10] en el año 2001. La segunda técnica aplicada es support vector machine (SVM), introducida por Cortes and Vapnik[2] en el año 1995. Se obtuvieron buenos resultados con ambas técnicas, siendo significativamente superior el resultado obtenido utilizando CRF con una disminución del error del 31,5 %.

Palabras clave: aprendizaje automático, crf, svm

1. Introducción

Dado un texto sin la presencia de acentos ortográficos, el problema de restauración automática de acentos ortográficos consiste en insertar acentos ortográficos en los lugares del texto en el que son requeridos por las normas de acentuación. Los acentos ortográficos son muy importantes para determinar el significado de una oración, sin embargo es muy común omitir este tipo de signo ortográfico en la escritura informal, e.g. es muy común omitir acentos ortográficos en letras mayúsculas.

La acentuación ortográfica en algunas palabras del idioma español sirve solamente como ayuda a la pronunciación. Sin embargo, en otras palabras la acentuación ortográfica elimina la ambigüedad de una oración. Tomando esto en cuenta, podemos clasificar las palabras con acento ortográfico del idioma español de la siguiente manera[3]:

1. Palabras sin ambigüedad. Existe una única forma correcta de escribir estas palabras, e.g. *acentuación* siempre debe ser escrita con tilde.

2. Palabras con ambigüedad. Las palabras pertenecientes a esta clase pueden cambiar su significado dependiendo si son escritas con o sin acentuación ortográfica. Por ejemplo: diferentes conjugaciones del mismo verbo (*canto/cantó, hable/hablé*), sustantivos (*papa/papá, secretaria/secretaría*), etc.

La restauración de acentos ortográficos es trivial para las palabras *sin ambigüedad*. Sin embargo, la restauración de acentos ortográficos en las palabras *con ambigüedad* requiere examinar el contexto de dicha palabra.

El problema de la restauración de acentos ortográficos se encuentra estrechamente relacionado con los problemas de desambiguación léxica; involucra aspectos de la desambiguación del significado de una palabra y aspectos del etiquetado gramatical[1]. Esta clase de problemas de desambiguación contiene un grupo de problemas muy similares entre sí, e.g. el problema de restauración de mayúsculas. En el problema de restauración de mayúsculas, al igual que en la restauración de acentos ortográficos, se intenta distinguir entre conceptos semánticos diferentes, e.g. papa (tubérculo) y Papa (obispo de Roma).

La restauración de acentos ortográficos es una problemática que ha sido atacada utilizando técnicas. Crandall[3] propone un método híbrido combinando un método basado en HMM y un modelo Bayesiano. Crandall utiliza ambos modelos eligiendo para cada palabra, uno u otro dependiendo de la confianza ofrecida por cada método. Simard[17], para la restauración de acentos ortográficos de textos en francés, también propone un método basado HMM combinado con un análisis morfo-sintáctico del corpus. Yarowsky[21,20] experimentó con métodos basados en HMM, clasificadores Bayesianos, y listas de decisión para la restauración de acentos ortográficos de textos en español y francés. Todos estos trabajos basan sus métodos de aprendizaje a nivel de palabras, Mihalcea[14] presenta un enfoque innovador al utilizar árboles de decisión a nivel de letras para restaurar acentos ortográficos en idioma rumano. La gran mayoría de estos trabajos reportan un porcentaje de éxito muy elevado, pero debe tenerse en cuenta el porcentaje de éxito del clasificador de línea base también es muy elevado. Por ejemplo, en el idioma francés aproximadamente el 85 % de las palabras no llevan acento ortográfico, y la forma correcta de más de la mitad de las palabras restantes puede deducirse determinísticamente. Esto deja una tasa base de éxito del 95 %[17].

En este trabajo se presenta la aplicación de dos técnicas de aprendizaje automático, CRF y SVM, para la resolución del problema de restauración automática de acentos ortográficos en adverbios interrogativos. Los adverbios interrogativos tienen una gran dependencia con el contexto en el que aparecen y presentan un problema particularmente difícil de resolver debido a su alto nivel ambigüedad.

El trabajo está organizado de la siguiente forma. En la siguiente sección se presenta una descripción del problema. La sección 3 presenta la técnica CRF, la primer técnica aplicada. Luego, en la sección 4 es introducido SVM, la segunda técnica que fue aplicada al problema. La discusión del análisis experimental y de los resultados obtenidos es presentada en la sección 5, mientras que las

[1] En inglés: *Part-of-speech tagging.*

conclusiones y las posibles líneas de trabajo futuro son presentadas en la sección 6.

2. Descripción del Problema

Un adverbio se define de la siguiente forma[5].

1. m. *Gram.* Palabra invariable cuya función consiste en complementar la significación del verbo, de un adjetivo, de otro adverbio y de ciertas secuencias. Hay adverbios de lugar, como *aquí, delante, lejos*; de tiempo, como *hoy, mientras, nunca*; de modo, como *bien, despacio, fácilmente*; de cantidad o grado, como *bastante, mucho, muy*; de orden, como *primeramente*; de afirmación, como *sí*; de negación, como *no*; de duda o dubitativos, como *acaso*; de adición, como *además, incluso, también*; de exclusión, como *exclusive, salvo, tampoco.* Algunos pertenecen a varias clases.
2. m. *Gram.* Los adverbios *como, cuando, cuanto y donde* pueden funcionar como relativos correspondientes a los adverbios demostrativos *así, según, tal, entonces, ahora, tan, tanto, aquí, allí,* etc.; pueden tener antecedente expreso o implícito; p. ej., *la ciudad donde nací; iré donde tú vayas.*
3. m. *Gram.* Pueden también funcionar como interrogativos o exclamativos. ORTOGR. Escr. con acento. *¿Cómo estás? ¡Cuánto lo siento!*

Los adverbios *como, cuando, cuanto y donde* pueden funcionar de forma interrogativa. Cuando un adverbio es interrogativo debe ser acentuado ortográficamente *cómo, cuándo, cuánto y dónde.* Además, el adverbio interrogativo puede formularse de forma directa o indirecta[18]. Un ejemplo de un adverbio interrogativo directo es el siguiente: *¿Adónde os marcháis?.* Una frase similar formulada como adverbio interrogativo indirecto es: *Dime adónde saldréis.*

Si bien la existencia de signos de interrogación es un fuerte indicador de la presencia de adverbios interrogativos directos, no existe un indicador claro que marque la presencia de los adverbios interrogativos indirectos. Además, dentro de una frase que contiene un adverbio interrogativo pueden presentarse adverbios no interrogativos, e.g. *Cabe preguntarse entonces, ¿por qué algunas enfermedades de origen vírico, como los catarros o la gripe, pueden sufrirse en repetidas ocasiones?.*

El problema que se ataca en este trabajo es el siguiente. Dado un texto del que fueron quitados todos los acentos ortográficos de sus adverbios: *cuándo, cuánto, dónde, cómo, adónde, qué*; se deberá clasificar cada palabra del texto en una de las siguientes clases:

- `O`. Toda palabra que no es uno de los adverbios considerados.
- `SIN_TILDE`. Si se trata de un adverbio no interrogativo.
- `CON_TILDE`. Si se trata de un adverbio interrogativo (potencialmente un adverbio exclamativo).

A continuación se presentan las dos técnicas de aprendizaje automático aplicadas a la resolución del problema.

3. Conditional Random Fields (CRF)

Conditional random fields[10] es un modelo estocástico para el etiquetado y la segmentación de datos secuenciales. Utiliza un modelo que define una probabilidad condicional $p(Y|x)$ sobre secuencias de etiquetas dada la observación de una secuencia particular, x. Para etiquetar una nueva secuencia observada x_*, el modelo condicional selecciona la etiqueta de la secuencia y_* que maximiza la probabilidad condicional $p(y_*|x_*)$. Un modelo CRF puede representarse como un grafo no dirigido en el que cada vértice representa una variable aleatoria y cada arista indica una dependencia entre las variables de los vértices que conecta. De esta manera, sea X la variable aleatoria que representa las secuencias de observación, definimos $G = (V, E)$ un grafo no dirigido en el que existe un nodo $v \in V$ por cada variable aleatoria representada por el elemento Y_v de Y.

La estructura del grafo que representa el modelo está dada por un único nodo X, que representa la entrada completa, y los nodos que representan los elementos Y forman una cadena de primer orden simple (ver la figura 1).

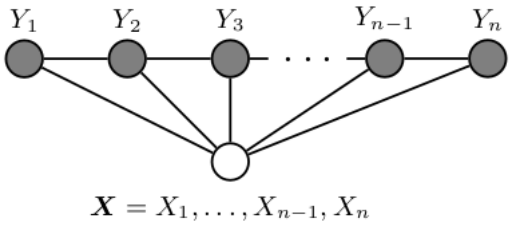

Figura 1. Estructura de un modelo CRF.

El modelo CRF resulta superior a otros métodos utilizados para el etiquetado de datos secuenciales, como Hidden Markov Model (HMM) o Maximum Entropy Markov Model (MEMM), debido a la relajación en el supuesto de independencia y debido a que evita el problema de sesgo de etiquetado presentados por estos modelos[19].

4. Support Vector Machines (SVM)

Support vector machines[8] es un clasificador lineal. Dado una serie de vectores de entrada, SVM intenta dividir linealmente el espacio vectorial creando una serie de regiones, a cada una de las cuales se le asocia una clase de salida. En el caso más simple de dos dimensiones tendremos un conjunto de instancias de entrenamiento $(x_1, y_1), ..., (x_N, y_N)$, siendo cada instancia x_i un vector en $\Re^N$ y siendo $y_i \in \{-1, +1\}$ la etiqueta de clase correspondiente. El algoritmo SVM aprende un hiperplano lineal que separa el conjunto de ejemplos positivos del conjunto de ejemplos negativos con *margen* máximo. El *margen* se define como

la distancia del hiperplano al punto más cercano de cada uno de los conjuntos de ejemplos.

El separador lineal se encuentra definido por un vector de pesos w y un sesgo b que representa la distancia del hiperplano al origen. La regla de clasificación de SVM es entonces $f(x, w, b) = \langle x \cdot w \rangle + b$ (ver la figura 2) [12].

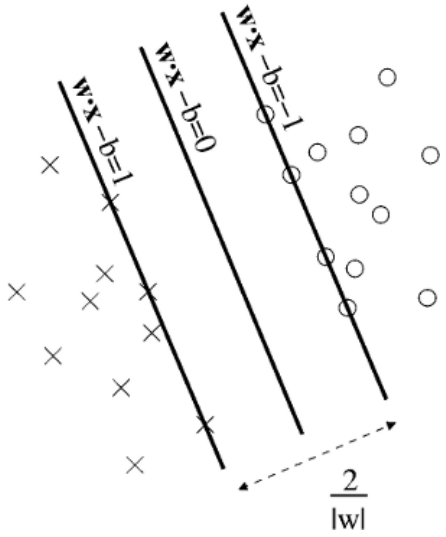

Figura 2. Hiperplano que maximiza el margen.

5. Evaluación de las técnicas propuestas

5.1. Métricas de Desempeño

Los clasificadores propuestos serán evaluados utilizando las métricas de *Precisión*, *Recall* y *Medida-F*. Durante la clasificación de elementos en una clase, el objetivo del clasificador consiste en encontrar la mayor cantidad de elementos de la clase buscada y al mismo tiempo confundir la menor cantidad de elementos de otras clases con los elementos de la clase buscada. La primera parte del objetivo anterior corresponde al concepto de *Recall*, mientras que la segunda parte corresponde al concepto de *Precisión*.

Generalmente las métricas de *Precisión* y *Recall* resultan en conflicto en el sentido de que cuando se intenta aumentar la cantidad de elementos hallados de una clase en particular (i.e. aumentar el *Recall*), usualmente se incrementa también la cantidad de errores cometidos al confundir elementos de otras clases (i.e. disminuye la *Precisión*) [16].

Para definir totalmente las métricas de *Precisión* y *Recall* debemos antes definir los siguientes conceptos.

- *Positivos Verdaderos* (PV) son elementos de la clase buscada que fueron correctamente identificados.
- *Negativos Verdaderos* (NV) son elementos que no pertenecen a la clase buscada que fueron correctamente ignorados y clasificados en una clase diferente.
- *Falsos Positivos* (FP), o errores de Tipo I, son elementos que pertenecen a otra clase y que fueron incorrectamente clasificados en la clase buscada.
- *Falsos Negativos* (FN), o errores de Tipo II, son elementos que pertenecen a clase buscada y que fueron incorrectamente clasificados en otra clase.

En base a estos conceptos podemos formular matemáticamente la métrica de *Precisión* como se muestra en la ecuación 1 y la métrica de *Recall* como se muestra en la ecuación 2 [1].

$$P = \frac{PV}{PV + FP} \tag{1}$$

$$R = \frac{PV}{PV + FN} \tag{2}$$

La Medida-F combina *Precisión* y *Recall* para obtener una única métrica definida como la media armónica ponderada de *Precisión* y *Recall*. En su formulación más general, la Medida-F puede expresarse como se presenta en la ecuación 3.

$$F_\alpha = \frac{P \times R}{(1-\alpha)P + (\alpha)R} \tag{3}$$

Generalmente se utiliza un valor de $\alpha = 0{,}5$ por lo tanto la ecuación 3 se reduce a la ecuación 4[11].

$$F_{0,5} = \frac{2 \times P \times R}{P + R} \tag{4}$$

5.2. Corpus Considerado

Para la evaluación de las técnicas elegidas se definió un corpus sobre el cual se realizó el entrenamiento y la validación de los clasificadores. El corpus utilizado se construyó en base a la unión de los corpus CESS Treebanks y CoNLL 2002. Para la construcción del corpus se utilizó la herramienta Natural Languaje Toolkit (NLTK) 2.0 [2]. Ambos corpus se encuentran incluidos y tokenizados en la herramienta NLTK, por lo tanto la utilización de NLTK facilitó la construcción del corpus de trabajo.

El corpus construido consta de un total de 562136 tokens, de los cuales 18677 son adverbios no interrogativos y 238 son adverbios interrogativos. En el cuadro 1 se muestra la proporción de las etiquetas asignas al corpus construido.

La gran mayoría de los tokens del corpus no son adverbios, un $96,64\,\%$, y por lo tanto les fueron asignados la etiqueta `O`. Esto nos asegura que utilizando

[2] Disponible para descargar en `http://www.nltk.org/`

Cuadro 1. Etiquetas asignadas al corpus utilizado.

Etiqueta	Cantidad de tokens	Proporción del corpus
O	543221	96,64 %
SIN_TILDE	18677	3,32 %
CON_TILDE	238	0,04 %

un clasificador de línea base muy simple podemos obtener fácilmente un $96,64\,\%$ de éxito asignando la etiqueta O a todos los tokens de nuestro corpus. Aún más, resulta trivial determinar si un token es un adverbio del tipo buscado, por lo tanto basta con asignar la etiqueta SIN_TILDE a todos los adverbios considerados para clasificar correctamente el $98,70\,\%$ de los adverbios y obtener un $99,96\,\%$ de éxito en nuestro clasificador.

5.3. Herramientas de Implementación

El clasificador basado en CRF fue implementado utilizando la herramienta MALLET 2.0.6, esta herramienta fue desarrollada por Andrew McCallum et al.[13]. MALLET se encuentra implementado en lenguaje Java y es una herramienta para el procesamiento estadístico del lenguaje natural que incluye herramientas para el etiquetado de secuencias mediante algoritmos como HMM, MEMM, y CRF.

Para la implementación del clasificador basado en SVM se utilizó la implementación SVM^{light}, desarrollada en lenguaje C por Thorsten Joachims[9]. Junto con SVM^{light} se utilizó la herramienta SVMTool, desarrollada por el grupo de Procesamiento de Lenguaje Natural de la Universidad Politécnica de Cataluña[6][7]. SVMTool es un generador de etiquetadores de secuencias para SVM^{light} implementado en lenguaje Perl, que brinda herramientas que facilitan la utilización de SVM^{light} en problemas de etiquetado de secuencias.

5.4. Clasificación Utilizando CRF

Como ya fue mencionado, se utilizó MALLET para la clasificación mediante CRF. Para definir las características de clasificación, MALLET provee un conjunto de *tubos* que pueden ser aplicados a los tokens del corpus para generar las características deseadas para la clasificación. Estos *tubos* permiten analizar los tokens mediante expresiones regulares, determinar la posición del token en la oración, obtener subconjuntos de caracteres del token, realizar conjunciones de características de diferentes tokens, etc.

La mayoría de las características utilizadas para la clasificación pudieron ser especificadas con el conjunto de *tubos* incluidos en MALLET, sin embargo para un pequeño conjunto de características fue necesario implementar *tubos* adicionales.

Cuadro 2. Características utilizadas para la clasificación mediante CRF.

Característica	Descripción
ADVERBIO	Tokens que son adverbios.
NOADVERBIO	Tokens que no son adverbios.
CAPITALIZED	Tokens cuya primera letra es mayúscula.
FIRST	Tokens que aparecen primeros en una oración.
BEGINNING	Adverbios que aparecen en primer o segundo lugar en una oración.
SIGN-QE	Tokens que representan signos de exclamación.
IN-QE	Adverbios en oraciones donde existen ocurrencias de signos de interrogación o exclamación.
ADVERBIO-QE	Adverbios seguidos o precedidos por un signo interrogativo o exclamativo.
PREV-SINT	Los siguientes tokens → *, lo la el los "*
NEXT-SINT	Los siguientes tokens → *el la los en las ha "*
ADVERBIO-SINT	Adverbios precedidos por PREV-SINT o sucedidos por NEXT-SINT.
PREV-CINT	Los siguientes tokens → *por - sobre ver a saber sé*
NEXT-CINT	Los siguientes tokens → *es le significa*
ADVERBIO-CINT	Adverbios precedidos por PREV-CINT o sucedidos por NEXT-CINT.
Conjunciones	Conjunción de todas las características del token anterior y el token actual, y todas las características del token actual y el token siguiente.

En el cuadro 2 se muestran las características utilizadas para la clasificación mediante CRF.

Las características PREV-SINT, NEXT-SINT, PREV-CONT y NEXT-CONT fueron incluidas luego de realizar un estudio sobre los tokens que preceden y que suceden a los adverbios interrogativos y no interrogativos en el corpus. Existen tokens que preceden o suceden a un adverbio que tienen una gran cantidad de ocurrencias sin importar si el adverbio es interrogativo o no. Sin embargo, otros tokens ocurren (en su mayoría) en el contexto de adverbios interrogativos o adverbios no interrogativos, pero no en ambos.

Se agregó la característica PREV-SINT a los tokens que preceden, con mayor ocurrencia, a adverbios no interrogativos y que a su vez no preceden a adverbios interrogativos. De la misma manera se agregó la característica NEXT-SINT a los tokens que suceden, con mayor frecuencia, a adverbio no interrogativo, pero no suceden a adverbios interrogativos.

De la misma manera, se establecieron las características `PREV-CONT` y `NEXT-CONT` para los token que preceden y suceden a adverbios interrogativos, pero no lo hacen con adverbios no interrogativos.

Para el entrenamiento se utilizó un modelo CRF de orden-1 totalmente conectado, y se utilizó un entrenador de verosimilitud por etiqueta. Por último, se prohibió la transición de etiquetas `CON_TILDE` a `CON_TILDE` ya que no pueden ocurrir dos adverbios interrogativos juntos en una oración.

5.5. Clasificación Utilizando SVM

Para la clasificación mediante SVM se utilizó la herramienta SVMTool. Se ha aplicado SVMTool a problemas de etiquetado gramatical en español e inglés, logrando en español un porcentaje de éxito en la clasificación del $96,89\,\%$[6].

Por defecto, la herramienta SVMTool utiliza para la clasificación las características definidas y optimizadas por Giménez et al.[6]. En este trabajo se tomaron como base las características defecto, pero se calibraron estas características para mejorar su desempeño en el problema de restauración de acentos ortográficos.

En el cuadro 3 se presentan las características de SVMTool utilizadas para la clasificación.

Cuadro 3. Características utilizadas en el clasificador SVM.

Característica	Descripción
`Tokens`	t_{-2}, t_{-1}, t_0, t_{+1}, t_{+2}
`Etiquetas`	e_{-2}, e_{-1}
`Bigramas`$_t$	(t_{-2}, t_{-1}), (t_{-1}, t_0), (t_{-1}, t_{+1}), (t_0, t_{+1}), (t_{+1}, t_{+2})
`Bigramas`$_e$	(e_{-2}, e_{-1}), (e_{-1}, a_{+1}), (a_{+1}, a_{+2})
`Trigramas`$_t$	(t_{-2}, t_{-1}, t_0), (t_{-2}, t_{-1}, t_{+1}), (t_{-1}, t_0, t_{+1}), (t_{-1}, t_{+1}, t_{+2}), (t_0, t_{+1}, t_{+2})
`Trigramas`$_e$	(e_{-2}, e_{-1}, a_{+1}), (e_{-1}, a_{+1}, a_{+2})
`SA`	Tokens cuya primera letra es mayúscula.
`aa`	Tokens en los que todas sus letras son minúsculas.
`Oración`	Puntuación de la oración $\rightarrow$ *. ? !*

Para la clasificación se utiliza una ventana de 5 tokens centrada en el token a etiquetar. Así t_0 representa el token a ser etiquetado, t_{+1} el token que sucede a t_0 en la oración, y t_{-1} el token que lo precede. La característica `Tokens` representa los unigramas formados por los tokens de la oración. De forma similar la característica `Etiquetas` representa las etiquetas de los tokens en la oración; esta

característica toma en cuenta las etiquetas de los dos tokens inmediatamente anteriores al token actual.

Además de los unigramas, para la clasificación se toman en cuenta bigramas y trigramas tanto de tokens (i.e. $\texttt{Bigramas}_t$, $\texttt{Trigramas}_t$) como de etiquetas (i.e. $\texttt{Bigramas}_e$ y $\texttt{Trigramas}_e$). Para el caso de los bigramas y trigramas de etiquetas ocurre una situación particular, para formar algunos n-gramas se utilizan etiquetas de tokens que se encuentran a la derecha del token que está siendo etiquetado, es decir, etiquetas que aún no han sido asignadas. Existen diferentes formas de resolver este problema, Nakagawa et al.[15] propone realizar un etiquetado en dos pasadas para conocer las etiquetas a izquierda y derecha de cada token. La herramienta SVMTool implementa la solución propuesta por Daelemans et al.[4] que consiste en utilizar etiquetas de ambigüedad que representan las posibles etiquetas de los tokens no clasificados.

Las característica `Oración` representa información de puntuación de la oración, e.g. si la oración finaliza con el token *'.'*, o *'?'*, o *'!'*.

En SVMTool existen diferentes modelos para realizar el entrenamiento del clasificador. En este trabajo se utiliza el *Modelo 0* de SVMTool con dos pasadas combinadas (`LRL`). Tal como lo mencionamos antes, en el *Modelo 0* se consideran clases de ambigüedad para el contexto desconocido. En un entrenamiento de dos pasadas primero se realiza un entrenamiento de izquierda a derecha (`LR`) y luego de derecha a izquierda (`RL`). Al momento de la clasificación ambas direcciones son evaluadas y es elegida la etiqueta que presenta mayor confiabilidad.

Por último, en SVMTool existen dos posibles esquemas de clasificación, el esquema *Goloso* y el esquema *Por oración*. En el esquema *Goloso* a cada token le es asignada la etiqueta que maximiza la función de puntuación del propio token. En el esquema *Por oración* a cada token le es asignada la etiqueta que maximiza la sumatoria de las funciones de puntuación de todos los tokens de la oración. Se realizaron experimentos con ambos esquemas de clasificación y el esquema *Por oración* arrojó mejores resultados por lo que se decidió utilizar este esquema.

5.6. Resultados Experimentales

La plataforma de hardware utilizada para los experimentos consiste en un computador con procesador Intel i3 de 2,13 GHz y 4 GB de RAM. Para la evaluación de los clasificadores se dividió el corpus por oraciones en 10 partes de tamaño similar; en el cuadro 4 pueden verse en detalle cada una de las partes.

Para cada clasificador se realizaron 10 entrenamientos diferentes, variando el conjunto de entrenamiento compuesto por 9 de las 10 partes del corpus y validando el resultado con la parte del corpus restante. En el cuadro 5 puede verse que el tiempo de entrenamiento y evaluación de cada conjunto de partes resultó similar en ambos clasificadores, pero durante el entrenamiento el clasificador CRF demandó un consumo de memoria RAM sensiblemente superior al clasificador SVM. El consumo de memoria fue una limitante importante durante el entrenamiento ya que agregando indiscriminadamente características se

Cuadro 4. Partes del corpus.

Parte	Tokens	SIN_TILDE	CON_TILDE
0	67144	2224	17
1	61651	2328	47
2	47137	1553	65
3	51632	1829	38
4	51255	1720	6
5	61023	1975	14
6	58352	1923	15
7	56056	1779	10
8	48725	1367	12
9	59161	1979	14

llegó fácilmente a consumir los 4 GB de RAM disponibles en la plataforma de hardware utilizada.

Cuadro 5. Requerimientos de cómputo.

Clasificador	T. entrenamiento	T. evaluación	Memoria
CRF	8 min.	20 seg. (∼2300 tokens/s)	1.6 GB
SVM	5 min.	30 seg. (∼1500 tokens/s)	120.0 MB

En el cuadro 6 puede verse que ninguno de los clasificadores arroja errores al clasificar la etiqueta `O`, y el error introducido en la clasificación de la etiqueta `SIN_TILDE` es mínimo. Estas características son fundamentales para mejorar el resultado obtenido por el clasificador de línea base. En la clasificación de la etiqueta `CON_TILDE` el clasificador CRF resulta superior al clasificador SVM. En la clasificación de esta etiqueta ambos clasificadores presentan una desviación estándar del error similar, pero el clasificador CRF obtiene en promedio un $16,56\,\%$ menos de errores que el clasificador SVM.

Cuadro 6. Error promedio porcentual por etiqueta.

Clasificador	O	SIN_TILDE	CON_TILDE
Línea base	0,00±0,00 %	0,00±0,00 %	100,00±0,00 %
SVM	0,00±0,00 %	0,02±0,03 %	87,30±11,55 %
CRF	0,00±0,00 %	0,04±0,07 %	70,74±12,51 %

Al comparar el total de errores obtenidos por cada clasificador luego de las 10 evaluaciones cruzadas del corpus, podemos ver que el clasificador CRF obtiene una disminución del $31,47\,\%$ del error en la clasificación de adverbios con respecto al clasificador de línea base, mientras que el clasificador SVM obtiene una disminución de solamente el $18,14\,\%$ (ver cuadro 7).

Cuadro 7. Total de errores por etiqueta.

Clasificador	SIN_TILDE	CON_TILDE	**Total**	**Proporción**
Línea base	0	238	238	100,00 %
SVM	3	193	196	82,35 %
CRF	7	156	163	68,49 %

Las matrices de confusión de los clasificadores muestran los tipos de errores cometidos. Los cuadros 8 y 9 muestran que para ambos clasificadores la gran mayoría de errores son cometidos al clasificar adverbios interrogativos con etiquetas SIN_TILDE. El $98,98\,\%$ de los errores del clasificador SVM y $95,09\,\%$ del clasificador CRF son debido a este tipo de confusión. Además, los errores cometidos al clasificar adverbios no interrogativos se deben a que son etiquetados como adverbios interrogativos. Resulta sorprendente ver que el clasificador CRF cometió un error al clasificar un adverbio interrogativo con la etiqueta O.

Cuadro 8. Matriz de confusión del clasificador SVM.

	O	SIN_TILDE	CON_TILDE
O	486053	0	0
SIN_TILDE	0	16695	3
CON_TILDE	0	194	30

Cuadro 9. Matriz de confusión del clasificador CRF.

	O	SIN_TILDE	CON_TILDE
O	486052	0	0
SIN_TILDE	0	16691	7
CON_TILDE	1	155	68

Analizando en detalle algunos de los errores cometidos por los clasificadores encontramos oraciones como las que siguen.

- *Si de él careciéramos, ¿para qué unas tareas que requieren esfuerzo, dedicación, capacidad y que -además- no mejoran ninguna economía?*
- *¿Puede un país pedir sanciones contra otro antes de que se resuelva una apelación de este último sobre el contencioso objeto de la sanción?*
- *Acabará aceptándose ese factor como un nuevo índice de las economías nacionales?*
- *Es cierta esa versión cristiana y consoladora que asegura que los sacrificios no son inútiles?*

Podemos ver que los signos interrogativos inducen errores al contener adverbios no interrogativos.

En los cuadros 10 y 11 se evalúan los clasificadores utilizando las métricas de desempeño definidas en la sección 5.1. En el cuadro 10 se presenta el desempeño de los clasificadores por etiqueta, y en el cuadro 11 se presenta el desempeño de los clasificadores para cada uno de los adverbios.

Cuadro 10. Métricas por etiqueta.

	Precisión		**Recall**		$\mathbf{F}_{0,5}$	
Etiqueta	**SVM**	**CRF**	**SVM**	**CRF**	**SVM**	**CRF**
`O`	1.00	1.00	1.00	1.00	1.00	1.00
`SIN_TILDE`	0.99	0.99	1.00	1.00	0.99	1.00
`CON_TILDE`	0.95	0.93	0.18	0.34	0.31	0.50

Cuadro 11. Métricas por adverbio.

		Precisión		**Recall**		$\mathbf{F}_{0,5}$	
Adverbio	**Cantidad**	**SVM**	**CRF**	**SVM**	**CRF**	**SVM**	**CRF**
que	14227	0.99	0.99	1.00	1.00	1.00	1.00
como	1574	0.96	0.97	1.00	1.00	0.98	0.98
cuando	447	0.99	0.99	1.00	1.00	1.00	1.00
donde	392	0.96	0.96	1.00	1.00	0.98	0.98
cuanto	55	0.96	0.96	1.00	1.00	0.98	0.98
adonde	3	1.00	1.00	1.00	1.00	1.00	1.00
qué	132	0.94	0.94	0.23	0.45	0.37	0.61
cómo	69	1.00	0.89	0.13	0.24	0.23	0.37
dónde	17	1.00	1.00	0.12	0.12	0.21	0.21
cuándo	4	0.00	0.00	0.00	0.00	0.00	0.00
cuánto	2	0.00	0.00	0.00	0.00	0.00	0.00

Ambos clasificadores presentan una métrica de *Precisión* muy elevada en la clasificación de adverbios interrogativos. Esto indica que ambos clasificadores cometen una cantidad muy pequeña de errores de Tipo I y por lo tanto ofrecen una certeza muy elevada en la clasificación de adverbios interrogativos. Sin embargo, el clasificador basado en CRF presenta una métrica de *Recall* significativamente más elevada que el clasificador basado en SVM para la clasificación de adverbios interrogativos. Esto indica que el clasificador CRF comete una menor cantidad de errores de Tipo II y por lo tanto logra clasificar correctamente una mayor cantidad de adverbios interrogativos. Esta diferencia en el desempeño de los clasificadores puede verse reflejada en la métrica de *Medida-F*; el clasificador CRF presenta una métrica de *Medida-F* claramente superior al clasificador SVM.

6. Conclusión

En este trabajo se construyó un corpus para la restauración automática de acentos ortográficos en adverbios interrogativos, y se propusieron dos clasificadores para resolver dicho problema. Ambos clasificadores obtuvieron buenos resultados, disminuyendo en $18,14\,\%$ y $31,47\,\%$ el error obtenido por el clasificador de línea base. El clasificador basado en CRF obtiene, comparativamente, una cantidad de aciertos sensiblemente superior al clasificador basado en SVM.

En ambos clasificadores se logró mantener al mínimo el error de clasificación de adverbios no interrogativos, entre $0,02 - 0,04\,\%$. Esto mantiene al mínimo la generación de falsos positivos, ofrece seguridad en la clasificación, y permite su combinación con otras técnicas para clasificar los adverbios interrogativos realizando sucesivas iteraciones.

Los tiempos de entrenamiento y clasificación resultaron más que aceptables, y permiten la utilización *en línea* de ambos clasificadores para la restauración de acentos ortográficos en tiempo real.

Por último, vale destacar que el clasificador basado en CRF resulta más claro gracias a su modelo estructurado; permite enfocarse en la definición de características del lenguaje y no es necesario definir decenas de n-gramas para simular una secuencia.

Referencias

1. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly, Beijing (2009)
2. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
3. Crandall, D.: Automatic accent restoration in spanish text (1995)
4. Daelemans, W., Zavrel, J., Berck, P., Gillis, S.: MBT: A memory-based part of speech tagger-generator. En: Ejerhed, E., Dagan, I. (eds.) Proceedings of the Fourth Workshop on Very Large Corpora. pp. 14–27 (1996)
5. Española, R.A.: Diccionario de la lengua española, vigésima segunda edición

6. Giménez, J., Màrquez, L.: SVMTool: A general pos tagger generator based on support vector machines. En: In Proceedings of the 4th International Conference on Language Resources and Evaluation. pp. 43–46 (2004)
7. Giménez, J., Màrquez, L.: SVMTool: Technical manual v1.3 (2006)
8. Joachims, T.: Making large-scale SVM learning practical. En: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge, MA, USA (1999)
9. Joachims, T.: SVMlight: Support Vector Machine (2008), University of Dortmund
10. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. En: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289. ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
11. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. En: Proceedings of the DARPA Broadcast News Workshop (1999)
12. Mata, F.C.: Etiquetado Estadístico de Roles Semánticos. Tésis doctoral, Universidad de Sevilla (2007)
13. McCallum, A.: MALLET: A machine learning for language toolkit (2002), `http://mallet.cs.umass.edu`
14. Mihalcea, R.: Diacritics restoration: Learning from letters versus learning from words. En: Gelbukh, A.F. (ed.) CICLing. Lecture Notes in Computer Science, vol. 2276, pp. 339–348. Springer (2002)
15. Nakagawa, T., Kudo, T., Matsumoto, Y.: Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines. En: Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (2001)
16. Raghavan, V., Bollmann, P., Jung, G.S.: A critical investigation of recall and precision as measures of retrieval system performance. ACM Trans. Inf. Syst. 7, 205–229 (1989)
17. Simard, M.: Automatic Insertion of Accents in French Text. En: Proceedings of the Conference on Empirical Methods in Natural Language Processing (1998), EMNLP-3
18. Veciana, R.: La acentuación española: Nuevo manual de las normas acentuales. Universidad de Cantabria (2004)
19. Wallach, H.M.: Conditional random fields: An introduction. Reporte Técnico MS-CIS-04-21, University of Pennsylvania, Philadelphia (2004)
20. Yarowsky, D.: A comparison of corpus-based techniques for restoring accents in spanish and french text. En: Proceedings, 2nd Annual Workshop on Very Large Corpora. pp. 19–32 (1994)
21. Yarowsky, D.: Decision lists for lexical ambiguity resolution: application to accent restoration in spanish and french. En: Proceedings of the 32nd annual meeting on Association for Computational Linguistics. pp. 88–95. ACL '94, Association for Computational Linguistics, Stroudsburg, PA, USA (1994)