Restauración automática de acentos ortográficos en adverbios interrogativos

Santiago Iturriaga, Diego Garat, y Guillermo Moncecchi

Facultad de Ingeniería,
Universidad de la República,
Montevideo, Uruguay
{siturria,dgarat,gmonce}@fing.edu.uy

Resumen La omisión de acentos ortográficos un error tipográfico muy frecuente en el idioma español; su restauración automática consiste en la inserción de acentos omitidos en los lugares que son necesarios. Los adverbios interrogativos son un caso especialmente dificultoso de este problema, ya que en muchas ocasiones no existen marcas claras que indiquen su presencia. Este trabajo presenta dos técnicas de aprendizaje automático, *Conditional Random Fields* (CRF) y *Support Vector Machines* (SVM), aplicadas a la resolución del problema de la restauración automática de acentos ortográficos para el caso específico de los adverbios interrogativos. Se obtuvieron buenos resultados con ambas técnicas, siendo sensiblemente superior el resultado obtenido utilizando un clasicador basado en CRF, y que utiliza como atributos los tokens que más comúnmente preceden y siguen a los adverbios interrogativos.

Palabras clave: crf, svm, restauración automática de acentos ortográficos

1. Introducción

Dado un texto sin la presencia de acentos ortográficos, el problema de su restauración automática 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, por ejemplo, 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. Podemos clasificar las palabras con acento ortográfico del idioma español de la siguiente manera [4]:

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 cada palabra. Este problema 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.

La restauración de acentos ortográficos es una problemática que ha sido atacada utilizando diferentes técnicas. Crandall [4] propone un método híbrido combinando un método basado en *Hidden Markov Models*(HMM) y un modelo Bayesiano. Simard [16], para la restauración de acentos ortográficos de textos en francés, también propone un método basado HMM combinado con un análisis morfosintáctico del corpus. Yarowsky [20,19] experimenta 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. Estos trabajos reportan un porcentaje de éxito muy elevado, pero debe tenerse en cuenta que el porcentaje de éxito 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 % [16].

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. Este tipo de adverbios tienen una gran dependencia con el contexto en el que aparecen y son un problema particularmente difícil de resolver debido a su alto nivel ambigüedad. Según nuestro conocimiento no existen antecedentes de trabajos previos orientados a resolver el problema planteado.

El trabajo está organizado de la siguiente forma. En la siguiente sección se presenta una descripción del problema y el corpus de trabajo. En la sección 3 se aborda el problema de restauración de acentos ortográficos como un problema de clasificación y se introducen los métodos CRF y SVM. La discusión del análisis experimental y de los resultados obtenidos es presentada en la sección 4, mientras que las conclusiones y las posibles líneas de trabajo futuro son presentadas en la sección 5.

2. Descripción del Problema

Los adverbios son palabras invariables que complementan el significado de un verbo, un adjetivo u de otro adverbio [6]. Algunos de ellos —*cuando, cuanto, como, ...*— pueden funcionar como relativos (*la ciudad donde nació*) o de forma interrogativa/exclamativa (*¿dónde nació?*), debiendo ser, en este último caso, acentuados ortográficamente.

Los adverbios interrogativos pueden formularse de forma directa o indirecta [17]. 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: *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* y *qué*, se debe 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 [1].

El corpus de trabajo se basa en la unión de los corpus CESS Treebanks y CoNLL 2002. Para la construcción del corpus se utiliza la herramienta Natural Languaje Toolkit (NLTK) 2.0 [2]. 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 asignadas al corpus construido.

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 %

La gran mayoría de los tokens del corpus (un $96,64\,\%$) no son adverbios, y por lo tanto se les asigna la etiqueta `O`. Esto nos asegura que utilizando un clasificador de línea base muy simple podemos obtener fácilmente un $96,64\,\%$ de precisión 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.

[1] Potencialmente podría tratarse de un adverbio exclamativo.
[2] Disponible para descargar en `http://www.nltk.org/`

3. Restauración de acentos como un problema de clasificación

En este trabajo se aborda el problema de la restauración de acentos ortográficos como un problema de clasificación. Con este propósito se introducen dos métodos utilizados para el reconocimiento de patrones: Conditional random fields y Support vector machines. A continuación se presentan en detalle cada uno de estos métodos.

3.1. Clasificación Utilizando CRF

Conditional random fields, propuesto por Lafferty et al. [11], 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. 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 [18].

En este trabajo, para la implementación del clasificador basado en CRF se utiliza la herramienta MALLET 2.0.6, una herramienta desarrollada en Java por Andrew McCallum et al. [13]. En el cuadro 2 se muestra un resumen de las características implementadas en MALLET y utilizadas para la clasificación mediante CRF.

Las características `PREV-SINT`, `NEXT-SINT`, `PREV-CONT` y `NEXT-CONT` se incluyen en el clasificador luego un estudio sobre los tokens que preceden y suceden a los adverbios del corpus construido para el problema. La característica `PREV-SINT` es asignada a los tokens que preceden con mayor frecuencia a adverbios no interrogativos, pero no preceden a adverbios interrogativos. La característica `NEXT-SINT` se agrega a los tokens que suceden con mayor frecuencia a adverbios no interrogativos, pero no preceden a adverbios interrogativos. De forma similar, se agregaron las características `PREV-CONT` y `NEXT-CONT` para adverbios interrogativos. Para determinar los tokens de estas características, se analiza por separado cada corpus de entrenamiento. En cada corpus de entrenamiento se seleccionan los tokens que cuplen cada una de las características mencionadas con mayor probabilidad, y se utilizan estos tokens para la definición de cada característica. De esta manera, nos aseguramos que las características utilizadas para cada evaluación son totalmente independientes del corpus utilizado para esa evaluación.

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 → e.g.: *, lo la el los "*
NEXT-SINT	Los siguientes tokens → e.g.: *el la los en las ha "*
ADVERBIO-SINT	Adverbios precedidos por PREV-SINT o sucedidos por NEXT-SINT.
PREV-CINT	Los siguientes tokens → e.g.: *por - sobre ver a saber sé*
NEXT-CINT	Los siguientes tokens → e.g.: *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 las del token actual y el token siguiente.

Para el entrenamiento se utiliza un modelo CRF de orden-1 totalmente conectado, y optimizando la verosimilitud por etiqueta. Adicionalmente, se prohíbe la transición de etiquetas CON_TILDE a CON_TILDE ya que no pueden ocurrir dos adverbios interrogativos juntos en una oración.

3.2. Clasificación Utilizando SVM

Support vector machines es un método de aprendizaje estadístico propuesto por Vapnik et al. [3]. Es sabido que SVM obtiene mejores resultados que métodos tradicionales —como redes neuronales— aún en grandes espacios dimensionales con pequeños conjuntos de entrenamiento; por esta razón ha sido aplicado con éxito a problemas de detección facial, detección y reconocimiento de objetos, reconocimiento de textos manuscritos, categorización, predicción, entre otros [2].

Support vector machines es un clasificador lineal: dada una serie de vectores de entrada, el método 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 $f(x, w, b) = \langle x \cdot w \rangle + b$.

Para la implementación del clasificador basado en SVM se utiliza la herramienta SVM^{light} desarrollada en lenguaje C por Thorsten Joachims [9,10]. Junto con SVM^{light} se utiliza SVMTool, una herramienta desarrollada por el grupo de Procesamiento de Lenguaje Natural de la Universidad Politécnica de Cataluña [7,8]. SVMTool es un generador de etiquetadores de secuencias para SVM^{light} implementado en lenguaje Perl que facilita la utilización de SVM^{light} en problemas de etiquetado de secuencias.

Por defecto, la herramienta SVMTool utiliza para la clasificación las características definidas y optimizadas por Giménez et al. En este trabajo se toman como base las características defecto, calibradas 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 basado en 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 → *. ? !*

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.

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 ngramas 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. [5] 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.

4. Evaluación

Los resultados de los clasificadores son evaluados utilizando las métricas de *Precisión*, *Recall* y *Medida-F* [1]:

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

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

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

Para la definición de las métricas presentadas deben definirse los siguientes coneptos:

- *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.

Para este trabajo se utiliza $F_{0,5}$ [12], por lo que la ecuación 3 se reduce a la ecuación 4.

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

Para la evaluación de los clasificadores se dividió el corpus por oraciones en 10 partes de tamaño similar; para cada clasificador se realizan diez entrenamientos diferentes, variando el conjunto de entrenamiento compuesto por nueve de las diez partes del corpus y validando el resultado con la parte del corpus restante.

Las matrices de confusión de los clasificadores muestran los tipos de errores cometidos. Los cuadros 4 y 5 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 basado en SVM y $95,09\,\%$ del clasificador basado en CRF son debido a este tipo de confusión. Además, los errores cometidos al clasificar adverbios no interrogativos se deben solamente a que son etiquetados como adverbios interrogativos.

Cuadro 4. Matriz de confusión del clasificador basado en SVM.

	O	SIN_TILDE	CON_TILDE
O	54322,1	0,0	0,0
SIN_TILDE	0,0	1867,4	0,3
CON_TILDE	0,0	19,3	4,5

Cuadro 5. Matriz de confusión del clasificador basado en CRF.

	O	SIN_TILDE	CON_TILDE
O	54322,1	0,0	0,0
SIN_TILDE	0,0	1867,7	0,8
CON_TILDE	0,1	16,0	7,7

A continuación se muestran algunos ejemplos de adverbios encontrados en el corpus junto con la etiqueta que el clasificador basado en CRF les asignará a cada uno de ellos.

- *Si de él careciéramos, ¿para qué/*`SIN_TILDE` *unas tareas que/*`SIN_TILDE` *requieren esfuerzo, dedicación, capacidad y que/*`SIN_TILDE` *—además— no mejoran ninguna economía?*
- *¿No se debate permanentemente —como/*`CON_TILDE` *toda religión y toda demencia— en el conflicto entre lo real y lo ficticio, lo percibido y lo proyectado, lo que/*`SIN_TILDE` *constriñe y lo que/*`SIN_TILDE` *exalta, los milagros y las bromas pesadas?*
- *-Que/*`CON_TILDE` *cómo/*`SIN_TILDE` *va a llamarse el chiquillo?*
- *En esta línea, Alberto Fernández se ha preguntado que/*`SIN_TILDE` *"si esto es lo que/*`SIN_TILDE` *ocurrió entonces cuando/*`SIN_TILDE` *CiU era influyente en Madrid, ¿qué/*`SIN_TILDE` *tiene que/*`SIN_TILDE` *pasar ahora con un escenario español totalmente diferente?"*
- *El dirigente popular ha evitado, no obstante, plantear su oferta de diálogo a CiU en el Parlamento en forma de ultimátum y ha asegurado que/*`SIN_TILDE` *el PP "no se precipitará" y que/*`SIN_TILDE` *espera a ver "qué/*`SIN_TILDE` *ficha mueve" la coalición que/*`SIN_TILDE` *lidera Jordi Pujol.*
- *Además, confirmó que/*`SIN_TILDE` *la marca de lujo Rolls Royce, adherida igualmente a BMW, permanecerá en el Reino Unido, por su añeja tradición británica y sus dimensiones, mucho menores que/*`SIN_TILDE` *Rover, aunque no precisó dónde/*`SIN_TILDE` *se instalará la nueva planta que/*`SIN_TILDE` *la fabricará.*
- *"Un experimento probará la preparación y las capacidades para el contexto militar del futuro, y sirve para que/*`SIN_TILDE` *veamos cómo/*`SIN_TILDE` *cada una de las fuerzas se desempeñará en una guerra".*

Podemos notar que las dos grandes causas de error en la clasificación son: por un lado los adverbios interrogativos indirectos, y por otro lado los adverbios no interrogativos contenidos en frases interrogativas.

En los cuadros 6 y 7 se evalúan los clasificadores utilizando las métricas de desempeño previamente definidas. En el cuadro 6 se presenta el desempeño de los clasificadores por etiqueta, y en el cuadro 7 se presenta el desempeño de los clasificadores para cada uno de los adverbios.

Cuadro 6. Métricas agrupadas 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`	**0.94**	0.91	0.18	**0.33**	0.31	**0.48**

Ambos clasificadores presentan una alta tasa de *Precisión*, lo que indica que se comete una cantidad muy pequeña de errores de Tipo I y por lo tanto ofrecen una certeza muy elevada. El clasificador basado en CRF presenta una tasa de *Recall* sensiblemente más elevada que el basado en SVM; esto indica que el clasificador basado en CRF comete una menor cantidad de errores de Tipo II y por lo tanto clasifica correctamente una mayor cantidad de casos.

Esta diferencia en el desempeño de los clasificadores puede verse reflejada en la métrica de *Medida-F*: el clasificador basado en CRF presenta un valor claramente superior al clasificador basado en SVM.

Cuadro 7. Métricas agrupadas por adverbio.

		Precisión		**Recall**		$\mathbf{F}_{0,5}$	
Adverbio	**Cantidad**	**SVM**	**CRF**	**SVM**	**CRF**	**SVM**	**CRF**
qué	142	**0.94**	0.92	0.23	**0.42**	0.36	**0.57**
cómo	72	**1.00**	0.89	0.14	**0.23**	0.24	**0.36**
dónde	18	0.67	**1.00**	0.11	**0.17**	0.19	**0.29**
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

En el cuadro 7 puede verse como los mejores resultados en la métrica *Recall* —y por ende también la métrica *Medida-F*— son obtenidos en la clasificación de los adverbios que cuentan con una mayor cantidad de ejemplos en el corpus. El adverbio *qué* siendo el adverbio con mayor ocurrencias en el corpus es a su vez es el adverbio para el que se obtienen los mejores resultados.

5. Conclusiones y trabajo futuro

En este trabajo se plantea el problema de la restauración automática de acentos ortográficos en adverbios interrogativos para el idioma español, un problema que —según nuestro conocimiento—- no ha sido atacado hasta el momento. Para su resolución se construye un corpus de aproximadamente 500k palabras mediante la unión de los corpus CESS Treebanks y CoNLL 2002. Debido a la falta de trabajos previos que aborden esta problemática, se proponen dos implementaciones su resolución: un clasificador basado en SVM y un clasificador basado en CRF. Los resultados obtenidos durante la evaluación muestran que la implementación del clasificador basado en CRF —que utiliza como atributos los tokens que más comúnmente preceden y siguen a los adverbios interrogativos— se comporta consistentemente mejor que la implementación del clasificador basado en SVM.

En la clasificación de adverbios no interrogativos la implementación del clasificador basado en CRF mantiene al mínimo el error de clasificación; minimizando la generación de falsos negativos, ofreciendo seguridad en la clasificación, y permitiendo la combinación de este método con otras técnicas mediante sucesivas iteraciones.

En la clasificación de adverbios interrogativos el clasificador basado en CRF mejora sensiblemente los resultados obtenidos al aumentar la cantidad de ejemplos en el corpus de entrenamiento. Esto parece indicar que para mejorar los resultados experimentales es necesario aumentar el tamaño del corpus considerado.

Las principales líneas de trabajo a futuro incluyen la construcción de un corpus de mayor porte, y la utilización de otras técnicas —como el etiquetado gramatical, el análisis morfosintáctico, etc.— para aumentar la información de contexto al momento de la clasificación.

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. Byun, H., Lee, S.W.: Applications of support vector machines for pattern recognition: A survey. En: Lee, S.W., Verri, A. (eds.) Pattern Recognition with Support Vector Machines, Lecture Notes in Computer Science, vol. 2388, pp. 571–591. Springer Berlin / Heidelberg (2002)
3. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
4. Crandall, D.: Automatic accent restoration in spanish text (1995)
5. 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)
6. Española, R.A.: Diccionario de la lengua española, vigésima segunda edición
7. 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)
8. Giménez, J., Màrquez, L.: SVMTool: Technical manual v1.3 (2006)
9. 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)
10. Joachims, T.: SVMlight: Support Vector Machine (2008), University of Dortmund
11. 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)
12. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. En: Proceedings,DARPA broadcast news workshop (1999)
13. McCallum, A.: MALLET: A machine learning for language toolkit. Disponible en `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. Simard, M.: Automatic Insertion of Accents in French Text. En: Proceedings of the Conference on Empirical Methods in Natural Language Processing (1998), EMNLP-3
17. Veciana, R.: La acentuación española: Nuevo manual de las normas acentuales. Universidad de Cantabria (2004)
18. Wallach, H.M.: Conditional random fields: An introduction. Reporte Técnico MS-CIS-04-21, University of Pennsylvania, Philadelphia (2004)
19. 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)
20. 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)