

Reentrenamiento: Aprendizaje Semisupervisado de los Sentidos de las Palabras*

Armando Suárez **Manuel Palomar**
Dep. de Lenguajes y Sistemas Informáticos
Universidad de Alicante
Aptd. 99, E-03080 Alicante, Spain
{armando, mpalomar}@dlsi.ua.es

German Rigau
Grupo IXA
Euskal Herriko Unibertsitatea
Donostia, Spain
rigau@si.ehu.es

Resumen: Este artículo presenta un algoritmo iterativo-incremental, *reentrenamiento*, que adquiere de forma automática nuevos ejemplos anotados semánticamente, asegurando una alta precisión. El algoritmo se inscribe dentro de los métodos de aprendizaje automático basados en corpus y usa los modelos de probabilidad de máxima entropía. Reentrenamiento consiste en la retroalimentación del corpus de entrenamiento, mediante sucesivos ciclos de aprendizaje y clasificación, de nuevos ejemplos clasificados con un grado alto de confianza. Este nuevo método se inspira en los algoritmos de coentrenamiento (*co-training*) pero asumiendo unas restricciones más fuertes a la hora de decidir qué ejemplos se etiquetan e incorporan a la siguiente iteración y cuáles no.

Palabras clave: desambiguación léxica, máxima entropía, basado en corpus, bootstrapping, co-training

Abstract: This paper presents *re-training*, a bootstrapping algorithm that automatically acquires semantically annotated data, ensuring high levels of precision. This algorithm uses a corpus-based system of word sense disambiguation that relies on maximum entropy probability models. The re-training method consists of the iterative feeding of training-classification cycles with new and high-confidence examples. The process relies on several filters that ensure the accuracy of the disambiguation by discarding uncertain classifications. This new method is inspired by *co-training* algorithms, but it makes stronger assumptions on when to assign a label to a linguistic context.

Keywords: Word Sense Disambiguation, Maximum Entropy, corpus-based, bootstrapping, co-training

1. Introducción

La resolución de la ambigüedad semántica de las palabras (WSD, *word sense disambiguation*) es un campo de desarrollo abierto dentro del procesamiento del lenguaje natural (PLN). La tarea consiste en asignar el sentido correcto a las palabras de entre las definiciones que se pueden encontrar en un diccionario electrónico. Es un problema difícil que genera gran interés entre la comunidad científica.

Actualmente hay dos aproximaciones metodológicas principales en este área: métodos basados en el conocimiento y métodos basa-

dos en corpus. La primera utiliza el conocimiento lingüístico previamente adquirido, y la segunda utiliza técnicas estadísticas y aprendizaje automático para inducir modelos del lenguaje a partir de grandes conjuntos de ejemplos textuales (Pedersen, 2001).

El aprendizaje automático y basado en corpus puede ser supervisado o no supervisado. Para el aprendizaje supervisado conocemos la clase de cada elemento dentro del conjunto de aprendizaje (en nuestro caso, la etiqueta de sentido) mientras que en el no supervisado la clasificación de los datos de entrenamiento no es conocida (Manning y Schütze, 1999).

En los últimos encuentros del *International Workshop on Evaluating Word Sense Disambiguation Systems*, SENSEVAL-2

* This paper has been partially supported by the Spanish Government (CICYT) under project number TIC2000-0664-C02-02 and the Valencia Government (OCyT) under project number CTIDIB-2002-151.

y SENSEVAL-3¹, entre otras, los investigadores mostraron los avances en el área que nos ocupa en la desambiguación de texto completo (*all-words task*) y de muestra léxica (*lexical-sample task*)². Los valores de eficacia conseguidos para estas tareas rondan, en general, el 70 %, por lo que podemos decir que queda aún mucho por hacer, más si tenemos en cuenta que WSD debería servir de apoyo a otras tareas del PLN como son recuperación de información, traducción automática o búsqueda de respuestas.

Existe un acuerdo más o menos amplio en que la falta de un corpus apropiado y suficiente grande representa un obstáculo para continuar progresando en este área. Es difícil conseguir un corpus anotado con sentidos para aprendizaje automático (Ng y Lee, 1996; Edmonds, 2000; Mihalcea, 2003), y los avances y esfuerzos recientes en su adquisición automática no hacen sino reforzar su importancia para este desarrollo crucial.

Los algoritmos iterativo-incrementales (*bootstrapping*) parecen una buena opción para esa adquisición automática de nuevos conjuntos de entrenamiento anotados. Básicamente, sólo se necesitan unos pocos ejemplos para iniciar un proceso iterativo que se retroalimenta, a partir de un conjunto no anotado, en sucesivos ciclos de aprendizaje y clasificación.

Co-training (Blum y Mitchell, 1998) es uno de estos algoritmos que se basa en dos vistas diferentes de los mismos datos de entrenamiento (por ejemplo, dos selecciones de atributos o *features* diferentes). Estas dos vistas son suficientes por sí mismas para clasificar eficazmente el conjunto de ejemplos no anotado. Estas vistas definen, por tanto, dos clasificadores simples: cada uno de ellos elige aquellas clasificaciones que considera más seguras para enriquecer el próximo entrenamiento del otro. Así, y de forma alternada, cada clasificador va procesando el resto de ejemplos que el otro no anota hasta que ya no quedan más ejemplos que clasificar. *Co-training* ha sido aplicado con éxito a la cla-

sificación de textos (Blum y Mitchell, 1998; Nigam y Ghani, 2000a), análisis estadístico (Steedman et al., 2003; Sarkar, 2001), adquisición de lexicones (Philips y Riloff, 2002), y reconocedores de entidades (Collins y Singer, 1999).

El problema principal que presenta este algoritmo es su rápida degradación de la precisión a partir de un determinado número de iteraciones (dependiendo del problema de clasificación y de los datos procesados). Algunos investigadores criticaron algunas de sus restricciones, como la necesidad de que las dos vistas sean totalmente independientes, al tiempo que propusieron modificaciones y mejoras (Abney, 2002; Collins y Singer, 1999).

Recientemente se ha publicado (Mihalcea, 2004) un estudio comparativo entre *Co-training* y *Self-training* y su aplicación a la tarea de WSD. La definición adoptada de este segundo método (parece que no hay un consenso, al menos, como lo hay para el primero) es que se realiza mediante un etiquetador que se reentrena con las anotaciones de sus propias clasificaciones anteriores.

En este artículo presentamos *Reentrenamiento*, un algoritmo iterativo-incremental que adquiere datos semánticamente etiquetados al tiempo que asegura altos niveles de precisión en la clasificación. El algoritmo usa un sistema de WSD supervisado basado en los modelos de máxima entropía (ME). En nuestro sistema, la información lingüística se representa en forma de vectores de atributos que identifican las ocurrencias de ciertos datos dentro de contextos que contienen ambigüedades lingüísticas. Entendemos por contexto al texto que acompaña a la ambigüedad y que es relevante para el propio proceso de desambiguación. Son usuales los atributos relacionados con palabras cercanas, lemas, categorías sintácticas, información de dominio, palabras clave, relaciones gramaticales, etc.

Este sistema se utiliza como núcleo de *Reentrenamiento*. Para cada sentido de una palabra se definen dos clasificadores ME débiles basados en diferentes conjuntos de atributos lingüísticos. La diferencia principal con *co-training* es que las dos vistas se usan en paralelo con el objetivo de obtener un consenso sobre qué etiqueta asignar a un contexto particular. Otros filtros adicionales permitirán, finalmente, incorporar algunos de estos contextos en el siguiente ciclo de entrenamiento.

Además, nuestro algoritmo no asume la

¹www.senseval.org

²La tarea de *texto completo* consistía en intentar desambiguar todas las palabras llenas (nombres, verbos, adjetivos y adverbios) que se encontraran dentro del texto propuesto, mientras que la *muestra léxica* trataba de la desambiguación de unas pocas palabras seleccionadas que aparecían como únicas instancias dentro de un conjunto más o menos extenso de contextos diferentes.

necesidad de preservar la distribución de sentidos en los nuevos conjuntos de entrenamiento, ya que nuestra intención es únicamente asegurar un alto grado de confianza en las clasificaciones, presumiendo que la fuente de ejemplos no anotados tiene una distribución desconocida.

En las siguientes secciones, se esbozan las principales características de nuestro sistema de WSD basado en ME y se describe el algoritmo de (Blum y Mitchell, 1998). A continuación, se detalla el algoritmo que hemos denominado *Reentrenamiento*, y se muestran los resultados de la evaluación de mismo. Finalmente, se exponen las conclusiones de este trabajo junto con una los desarrollos que se están llevando a cabo actualmente y los previstos para un futuro cercano.

2. El sistema de WSD basado en máxima entropía

El modelado con ME proporciona un marco para la integración de información para clasificación desde muchas fuentes heterogéneas (Manning y Schütze, 1999). Los modelos de probabilidad de ME han sido utilizados con éxito en tareas del PLN tales como POS tagging o detección de los límites de la frase (Ratnaparkhi, 1998).

El método de WSD usado para este trabajo está basado en los modelos de probabilidad condicional de ME, lo que ha resultado en una implementación de un método supervisado de aprendizaje automático que obtiene clasificadores de sentidos de palabras a partir de un corpus anotado. Se entiende como clasificador obtenido por esta técnica como un conjunto de coeficientes que se estiman mediante un algoritmo de optimización, cada uno asociado a un atributo (*feature*) observado en el corpus de entrenamiento. El principal objetivo es obtener una distribución de probabilidad que maximice la entropía, esto es, asumiendo la máxima ignorancia sobre los datos de entrenamiento de tal forma que no se induce ningún conocimiento que no esté propiamente en los datos.

Nuestro sistema (que denominaremos ME-WSD) se basa en una implementación propia en C++, cuyos detalles se pueden consultar en (Suárez y Palomar, 2002) y (Suárez, 2004).

2.1. Descripción de atributos

El conjunto de atributos definido para el entrenamiento del sistema se pueden consultar

- **O**: la forma en que está escrita la palabra objetivo
- **l**: lemas de palabras llenas en posiciones $\pm 1, \pm 2, \pm 3$ (“relaxed” definition)
- **o**: palabras en $\pm 1, \pm 2$,
- **s**: palabras en $\pm 1, \pm 2, \pm 3$
- **p**: categorías sintácticas en $\pm 1, \pm 2, \pm 3$
- **b**: lemas de composiciones de palabras en $(-2, -1), (-1, +1), (+1, +2)$
- **c**: composiciones de palabras en $(-2, -1), (-1, +1), (+1, +2)$
- **km**: lemas de nombres en cualquier posición y que coocurren al menos $m\%$ veces con un sentido concreto
- **r**: rol gramatical de la palabra objetivo
- **d**: la palabra de la que depende la palabra objetivo
- **v**: el verbo del que depende la palabra objetivo
- **m**: palabra compuesta de la que forma parte la palabra objetivo
- **W**: palabras llenas en $\pm 1, \pm 2, \pm 3$ (definición relajada)
- **L, S, B, C, P, M and D**: versiones “relajadas”

Figura 1: Atributos usados en el entrenamiento del sistema

tar en la figura 1 y está inspirado en los trabajos de (Ng y Lee, 1996) y (Escudero, Márquez, y Rigau, 2000). Los atributos se definen automáticamente y dependen de los datos que se encuentren en el corpus de aprendizaje. Básicamente se trata de palabras, combinaciones de palabras, categorías sintácticas, y en general, información que es posible manejar porque la proporciona un analizador (*parser*).

En realidad, cada ítem en la figura 1 agrupa a varios conjuntos de atributos. La mayoría de ellos dependen de las palabras en el contexto cercano (por ejemplo, *s* el conjunto de todos los atributos posibles definidos a partir de las palabras que se encuentran en cada ejemplo y en posiciones $w_{-3}, w_{-2}, w_{-1}, w_{+1}, w_{+2}, w_{+3}$ relativas a la palabra objetivo). Los tipos denominados mediante letras mayúsculas son redefiniciones buscando una menor carga computacional (los detalles pueden consultarse en (Suárez, 2004)).

Los atributos de palabras clave (*km*) se

basan en el trabajo de (Ng y Lee, 1996). Un filtrado de los nombres proporciona información de frecuencia de los mismos con sentidos concretos. Por ejemplo, supongamos que $m = 10$ para un conjunto de 100 ejemplos de *interés*: si el nombre *banco* aparece 10 veces o más, en cualquier posición relativa, entonces se define un atributo para esa palabra y esa clase.

Adicionalmente, haciendo uso de ciertas propiedades gramaticales, se han definido otros conjuntos de atributos: roles de las palabras objetivo (r) (sujeto, objeto, complemento, etc.) y dependencias dentro del árbol sintáctico que pueda proporcionar el analizador (d and D).

3. El método Reentrenamiento

El método que se va a proponer en esta sección se inscribe dentro del amplio abanico de técnicas incrementales o de semilla (*bootstrapping*) y, más concretamente, el punto de partida es el trabajo de (Blum y Mitchell, 1998), donde se propone un método de clasificación iterativo, el *co-training* (a partir de ahora **coentrenamiento**).

Sin embargo, uno de los primeros métodos incrementales que se citan específicamente para WSD es el de (Yarowsky, 1995). Se trata de un método no supervisado que se basa en dos restricciones: que una palabra tiende a tener un único sentido dentro de un mismo discurso, y también dentro de una misma “colocación” (*one sense per discourse*, *one sense per collocation*). El método se evaluó sobre un pequeño conjunto de palabras con dos posibles sentidos cada una. Partiendo de las definiciones de un diccionario, se construyó una semilla con colocaciones representativas de cada sentido y se utilizó como entrada para un algoritmo de listas de decisión. El incremento del corpus anotado se hacía con aquellas clasificaciones que superaban un cierto umbral de probabilidad.

Este trabajo tuvo, y tiene (Abney, 2004), un gran impacto en la comunidad científica especializada en WSD por la alta precisión conseguida (95% aproximadamente), aunque es evidente que las condiciones del experimento están un poco alejadas de la realidad. (Blum y Mitchell, 1998) afirmaron que este algoritmo es un caso particular de su propuesta.

3.1. Coentrenamiento

En el coentrenamiento, dos clasificadores sencillos (*weak learners*) pueden ayudarse el uno al otro a mejorar su acierto siempre y cuando concurren ciertas condiciones. Los dos clasificadores se entrenan a partir de un pequeño **conjunto anotado** (CA), la *semilla*, y clasifican un **conjunto no anotado** (CNA). De estas dos clasificaciones, cada uno elige los ejemplos que considera más fiables y los incorpora al conjunto anotado para volver a entrenar y clasificar en un proceso iterativo que termina según unos criterios preestablecidos.

A medida que se ejecutan las iteraciones, el CA se va haciendo mayor con las contribuciones de cada clasificador. Así el clasificador que llamaremos h_1 utiliza en la siguiente iteración los ejemplos que ha clasificado el segundo clasificador, h_2 , y viceversa. De esta forma se espera que se reduzca el error cometido por cada clasificador en una tasa significativa.

Los clasificadores son diferentes porque utilizan dos vistas distintas de los mismos datos para aprender. El término ‘vista’ podemos asimilarlo a una selección de atributos, es decir cada clasificador entrena con conjuntos distintos de atributos pero sobre los mismos ejemplos. El problema que (1998) mostraban era la clasificación de páginas web en dos posibles clases, páginas personales del personal docente y las que no lo son. Establecen, pues, una partición binaria de los datos de entrenamiento, dos únicas clases, ejemplos ‘positivos’ y ‘negativos’. Un clasificador aprendía a partir del conjunto de palabras que formaban el texto de la página y el otro del conjunto de palabras que se encontraban dentro de los enlaces de esa página hacia otras.

El coentrenamiento puede aplicarse a problemas de clasificación que cumplan las siguientes condiciones:

1. cada vista de los datos debe ser suficiente por si misma para realizar la tarea
2. los ejemplos anotados por coentrenamiento obtienen esa misma clase con cualquiera de las dos vistas
3. las vistas son condicionalmente independientes dada la clase

El algoritmo propuesto por (Blum y Mit-

chell, 1998) es el siguiente³:

```
CA: es el conjunto anotado (semilla)
    con dos únicas clases: positivo
    y negativo
CNA: es el conjunto no anotado
p: es la cantidad de anotaciones
    positivas por iteración y
    clasificador
n: es la de negativas

Crear un subconjunto CNA' de
    ejemplos seleccionados
    aleatoriamente del CNA

Para k iteraciones
    Entrenar un clasificador h1
        con CA (vista 1)
    Entrenar un clasificador h2
        con CA (vista 2)
    Etiquetar p ejemplos positivos
        y n negativos del CNA' con h1
    Etiquetar p ejemplos positivos
        y n negativos del CNA' con h2
    Añadir estos ejemplos al CA
    Rellenar CNA' con 2p + 2n ejemplos
        elegidos aleatoriamente de CNA
```

Por ser foco de atención y esperanza de muchos investigadores en métodos supervisados, ávidos de aprovechar una ingente cantidad de información no anotada, esta primera propuesta ha tenido bastantes revisiones y matizaciones ((Abney, 2002; Collins y Singer, 1999; Nigam y Ghani, 2000b)). Además de ser un método con unas restricciones muy fuertes, el problema del coentrenamiento radica en la rápida degradación del acierto cuando se acumulan los errores de clasificación.

3.2. Descripción de Reentrenamiento

En esta sección se describen los fundamentos de nuestro método semisupervisado. Su principal objetivo es obtener clasificaciones con un alto grado de confianza. Dado que nuestra intención es aplicarlo a WSD, hemos usado el sistema basado en ME-WSD descrito en la Sección 2.

³La decisión de no utilizar el CNA al completo es simplemente una cuestión de eficiencia, suponiendo que el tamaño del CNA es demasiado grande y ralentizaría en demasía la clasificación en cada iteración. Así, eligen aleatoriamente un subconjunto que se rellena en cada iteración manteniendo un tamaño constante. Se supone que no supone una merma del resultado final, aunque no aclaran el impacto de la selección aleatoria de la semilla.

Vamos a suponer una palabra w con tres sentidos, s_1 , s_2 y s_3 . Supongamos, así mismo, un corpus anotado (CA) que servirá como conjunto de ejemplos para el aprendizaje, y un corpus no anotado (CNA) que se ha de clasificar.

Un clasificador “normal” sería entrenado, vendría definido por un aprendizaje desde el CA y, posteriormente, sería utilizado para clasificar los contextos presentes en el CNA en las tres clases posibles, en cualquiera de los sentidos de la palabra w . Digámoslo así, se aprende todo de una vez y se obtiene el clasificador que clasifica todo el CNA en una única ejecución.

Podríamos plantearnos otra forma de trabajar, que en vez de un único clasificador, sucesivos clasificadores aprovecharan la información suministrada por el anterior, un proceso iterativo que fuera mejorando el clasificador poco a poco. Supongamos que aprendemos del CA pero sólo clasificamos una pequeña porción del CNA, no por completo. La intención de clasificar tan sólo unos cuantos contextos es que, por el criterio que sea, aseguremos que la precisión es muy alta, aunque la cobertura sea mínima.

Pero esta pequeña porción de contextos clasificados en el primer aprendizaje son añadidos al CA previo, y pasan a ser ejemplos de un corpus de aprendizaje mayor que será el material de un segundo entrenamiento. A este nuevo corpus, el CA más los pocos ejemplos obtenidos del CNA lo llamaremos CA2, y al CNA del que eliminamos justo esos contextos ya etiquetados en la primera clasificación lo llamaremos CNA-2.

Tras el segundo aprendizaje, el clasificador que obtenemos está “más informado”, tiene más ejemplos de los que aprender, el CA2. Si procesamos el CNA-2 con este segundo clasificador y seleccionamos unas cuantas clasificaciones más, podemos obtener de la misma forma que antes un CA3 y un CNA-3, y construir un tercer clasificador aún más informado.

Y así, hasta llegar a CA_n y CNA_n , donde incluso puede que el CNA esté vacío porque todos los ejemplos han sido clasificados y han pasado al CA.

Esta forma de trabajar se basa en la esperanza de que los sucesivos clasificadores son capaces de detectar los contextos cuyas etiquetas son muy fiables, y que gracias a que un clasificador es mejor que el anterior, cada

vez va a ser más fácil etiquetar los contextos que van quedando en los CNA, así hasta clasificarlos todos.

El problema, ya lo hemos dicho, es como filtrar qué etiquetas son fiables y cuáles no. Debemos encontrar un proceso que automáticamente rechace las clasificaciones de las que no se puede asegurar su corrección. Esta es la motivación de nuestro método, que pasamos a describir a continuación.

3.3. Un ejemplo

Sea la misma palabra w , con sus tres sentidos $s1$, $s2$ y $s3$, y el CA y el CNA que le son propios. Dividamos el aprendizaje en sentidos, esto es, construyamos tres clasificadores, cada uno correspondiente a un sentido: $c1$, $c2$ y $c3$. El objeto de cada uno de estos clasificadores es decidir si un contexto pertenece o no a su clase, es decir, $c1$ clasificará los contextos del CNA en “sí $s1$ ” y “no $s1$ ”, al igual que los otros dos con sus propios sentidos.

Este proceder necesita de un paso previo: reetiquetar el CA para cada clasificador. El corpus de aprendizaje del clasificador $c1$ a construir ha de contener ejemplos positivos y negativos del sentido $s1$. Fácilmente, todos los ejemplos anotados como $s1$ son positivos y los etiquetados como $s2$ o $s3$ son negativos. En realidad, “triplicamos” el CA, puesto que los otros dos sentidos tendrán sus propios conjuntos de entrenamiento con ejemplos positivos y negativos.

CA	CNA
$e_1.s1$	x_1
$e_2.s2$	x_2
$e_3.s3$	x_3

Corpus original

CA.s1	CA.s2	CA.s3	CNA
$e_1.SÍ$	$e_1.NO$	$e_1.NO$	x_1
$e_2.NO$	$e_2.SÍ$	$e_2.NO$	x_2
$e_3.NO$	$e_3.NO$	$e_3.SÍ$	x_3

Nuevo corpus

Supongamos que $e_1.s1$, $e_2.s2$ y $e_3.s3$ son los tres ejemplos del CA, cada uno etiquetado con un sentido, y que tres contextos del CNA, x_1 , x_2 y x_3 , tres contextos del CNA, son la entrada a los tres clasificadores y que el resultado obtenido es el siguiente:

	$c1$	$c2$	$c3$
x_1	SÍ	NO	SÍ
x_2	NO	NO	NO
x_3	SÍ	NO	NO

Consenso

El contexto x_1 ha sido clasificado como positivo por el clasificador del sentido $s1$ y por el del $s3$, el contexto x_2 no ha obtenido clasificaciones positivas en ninguno de los tres, y el tercero, x_3 sólo pertenece al sentido $s1$ por que sólo $c1$ lo ha etiquetado como positivo.

Un criterio de fiabilidad va a ser que sólo aquellos contextos de los que estamos seguros de su etiqueta son susceptibles de pasar del CNA al CA. En este caso, x_1 es ambiguo puesto que dos sentidos lo reclaman como “suyo”, y x_2 no pertenece a ninguno. Los tres clasificadores sí están de acuerdo en que x_3 es del sentido $s1$, hay consenso entre ellos.

El siguiente paso sería trasvasar los contextos consensuados del CNA al CA, pero tenemos tres CA distintos. La operación es obvia, x_3 pasa como positivo al corpus del sentido $s3$, y como negativo a los otros, eliminándolo del CNA. Los corpus de entrenamiento con un ejemplo más, y el no anotado con uno menos. El proceso se repite hasta que no se consigue clasificar positivamente ningún contexto más del CNA.

CA.s1	CA.s2	CA.s3	CNA
$e_1.SÍ$	$e_1.NO$	$e_1.NO$	x_1
$e_2.NO$	$e_2.SÍ$	$e_2.NO$	x_2
$e_3.NO$	$e_3.NO$	$e_3.SÍ$	
$x_3.SÍ$	$x_3.NO$	$x_3.NO$	

Trasvase

Complicemos un poco más el trabajo de los clasificadores. Hasta ahora no hemos dicho nada de la información utilizada en los aprendizajes binarios. Se entiende que para todos se ha elegido un conjunto de atributos con los que caracterizar los ejemplos. Supongamos que establecemos, no uno, sino dos clasificadores por sentido: $(c11, c12)$, $(c21, c22)$ y $(c31, c32)$, de tal forma que los clasificadores $cx1$ son entrenados con el conjunto de atributos A_1 , y los $cx2$ con el conjunto A_2 .

Por aclarar, y como ejemplo, supongamos que $A_1 = LWS$, las palabras y lemas alrede-

dor de w , y $A_2 = BC$, las composiciones de 2 palabras y lemas alrededor, también, de w .

La clasificación de los tres contextos se hace ahora en dos pasos: primero deben estar de acuerdo los dos clasificadores de cada sentido, y luego debe haber consenso entre los tres sentidos. El siguiente ejemplo comienza, otra vez, desde los corpus originales.

	c11	c12	c21	c22	c31	c32
x_1	SÍ	NO	SÍ	NO	SÍ	SÍ
x_2	NO	NO	SÍ	NO	NO	NO
x_3	SÍ	SÍ	NO	SÍ	NO	NO

Consenso entre clasificadores

Ahora cada sentido debe proponer sus positivos por acuerdo entre los dos clasificadores que trabajan para él. De esta forma, el contexto x_1 es positivo para el sentido $s3$ porque $c31$ y $c32$ están de acuerdo en ello, mientras que las otras parejas de clasificadores no lo han hecho. El resultado es que cada sentido propone sus positivos tal y como se muestra a continuación:

	c1	c2	c3
x_1	NO	NO	SÍ
x_2	NO	NO	NO
x_3	SÍ	NO	NO

Consenso entre sentidos

Ahora tenemos dos contextos clasificados, x_1 para $s3$, que pasará al corpus de aprendizaje como positivo para este sentido y como negativo para los otros, y x_3 para $s1$, que de igual forma será positivo en este sentido y negativo en los otros.

CA.s1	CA.s2	CA.s3	CNA
e_1 .SÍ	e_1 .NO	e_1 .NO	
e_2 .NO	e_2 .SÍ	e_2 .NO	x_2
e_3 .NO	e_3 .NO	e_3 .SÍ	
x_1 .NO	x_1 .NO	x_1 .SÍ	
x_3 .SÍ	x_3 .NO	x_3 .NO	

Trasvase

Este proceso se repite, nuevamente, hasta que no se consiguen nuevos ejemplos positivos para ningún sentido, o por cualquier otro criterio de parada. En realidad, el método que estamos proponiendo es más complejo

puesto que también se establecen valores de probabilidad mínimos para dar una respuesta positiva a un contexto, pero esto lo contamos ya en la siguiente sección.

El objetivo siempre es dificultar la clasificación errónea, y se espera de este sistema de acuerdos a dos niveles el asegurar la corrección de las etiquetas.

4. Método propuesto

Partiendo de estas ideas se va a proponer un método de semilla cuyo objetivo es obtener ejemplos de corpus no anotados con la máxima fiabilidad de etiquetado posible. A este método lo llamamos **Reentrenamiento**. Todo lo que a continuación se expone se entiende que se refiere a una palabra concreta.

Se parte de un corpus anotado (CA) y otro no anotado (CNA), ambos de una palabra con c clases (en nuestro caso, sentidos de WN).

Reentrenamiento se ha planteado como un proceso iterativo de incorporación de nuevos ejemplos al CA. La elección de estos nuevos ejemplos se hace aprendiendo del CA y clasificando el CNA.

El aprendizaje se divide en tantos aprendizajes binarios como clases tenga la palabra. Del CA se obtienen c conjuntos de entrenamiento, de forma que cada uno contiene ejemplos positivos y negativos, siendo positivos los anotados como pertenecientes a una clase y negativos los que son positivos para las otras clases.

De cada conjunto de entrenamiento, se obtienen 2 clasificadores binarios para cada clase utilizando conjuntos de atributos distintos (en total, tendremos $2 \times c$ clasificadores binarios) y se clasifica el CNA con ambos. De cada par de clasificaciones, se produce una **propuesta parcial**, una clasificación única por clase. Tendremos, pues, c propuestas parciales.

El siguiente paso consiste en comparar las propuestas binarias parciales para obtener una única **propuesta común** de nuevos ejemplos positivos. Se propone un consenso excluyente de tal forma que los ejemplos considerados positivos por más de una clase no se tienen en cuenta. La propuesta común consiste en todos aquellos ejemplos que sólo son positivos para una única clase.

Los ejemplos de la propuesta común se incorporan a su conjunto de entrenamiento

correspondiente y se reconstruyen los clasificadores binarios para volver a clasificar el CNA. Se repite el proceso hasta un número de iteraciones suficiente o predeterminado.

Dicho de otra forma, el reentrenamiento de una palabra se basa en varios entrenamientos parciales binarios, varias clasificaciones binarias y la combinación de éstas en una propuesta única que alimenta a la siguiente iteración, siguiendo el siguiente esquema procedural:

Generación de semillas: para cada clase de la palabra a procesar, se genera una semilla con ejemplos positivos y negativos, en principio utilizando todos los ejemplos disponibles en el CA. Estas semillas son los conjuntos de entrenamiento particulares de cada clase, y a los que se irán incorporando nuevos ejemplos positivos y negativos.

Entrenamientos binarios (propuestas parciales): para cada clase de la palabra, se activan dos entrenadores (programas de aprendizaje basados en máxima entropía y diferentes conjuntos de atributos) y se obtienen dos clasificaciones del CNA. Estos clasificadores, dos por sentido, trabajan con 2 clases únicamente, positivo y negativo (es sentido x , o no lo es).

A partir de estas dos clasificaciones, se realiza una propuesta parcial por consenso de incorporación de ejemplos para cada uno de los sentidos: para un sentido x , si sus dos clasificadores están de acuerdo en que un contexto es positivo éste se propone como ejemplo candidato. Cada sentido tendrá su propia lista de candidatos, su propuesta parcial.

Consenso entre sentidos (propuesta combinada): la elección de los ejemplos que se añadirán a los conjuntos de entrenamiento pasa por la verificación conjunta de todas las propuestas. Para cada candidato se comprueba que sólo una de las propuestas lo haya anotado como positivo; en caso contrario se entiende que el contexto es ambiguo.

Regeneración de los conjuntos de entrenamiento: a partir de la propuesta combinada, cada clase incorpora sus ejemplos positivos a su conjunto de en-

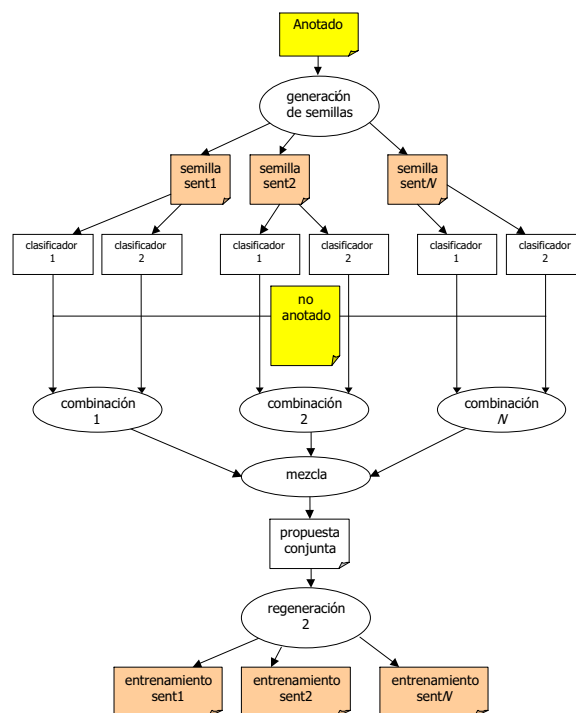


Figura 2: Esquema de Reentrenamiento

entrenamiento particular y el resto como negativos.

La siguiente iteración comienza en “entrenamientos binarios”, y el proceso finaliza en un número de iteraciones establecido por parámetro, o cuando ya no se consiguen nuevos ejemplos desde el CNA.

Una representación gráfica de este proceso puede verse en la figura 2.

5. Criterios de calidad de la clasificación

Se trata de establecer qué ejemplos del CNA se incorporan a los conjuntos de entrenamiento como positivos. Se definirán detalladamente los filtros apuntados anteriormente, los que se pretende que aseguren la correcta clasificación de algunos ejemplos, los que van a ser incorporados al posterior aprendizaje.

Para las propuestas parciales: en la fase de confección de las propuestas parciales, las que resultan del proceso de cada sentido, el primer criterio es que los dos clasificadores entrenados para un sentido en particular den como positivo un

ejemplo. Si alguno de los dos, o los dos, lo clasifican como negativo, no será propuesto.

Además, podemos establecer un **umbral** de confianza basado en la diferencia entre la probabilidad de que sea positivo y la probabilidad de que sea negativo. Supongamos:

$i = \{1|2\}$, índice de clasificador binario
 e : es un ejemplo no anotado
 u : es el umbral
 p_i : la probabilidad de que e sea positivo en el clasificador i
 n_i : la probabilidad de que e sea negativo en el clasificador i
 c : es la clase de los dos clasificadores binarios

e es positivo para la clase c
si $p_1 - n_1 > u$ y $p_2 - n_2 > u$

Para la propuesta común: después de que para cada sentido se propongan los candidatos a ser etiquetados, el primer paso para la confección de la propuesta combinada es comprobar los ejemplos que sólo son positivos en una de las propuestas parciales, los que sólo son “reclamados” por un único sentido. Los demás, aquellos ejemplos que son propuestos como pertenecientes a más de un sentido, son rechazados.

Una vez que se ha hecho esta combinación de las propuestas parciales, se eligen los ejemplos que obtienen el mayor valor de probabilidad dentro de su clase. Es como si, para cada clase, ordenáramos todos “sus” ejemplos, de mayor a menor probabilidad y sólo escogiéramos el primero (o los primeros si hay varios que obtienen la máxima probabilidad de su clase). Puesto que se han utilizado dos clasificadores para cada propuesta parcial, la suma de las probabilidades ($p_1 + p_2$) es el criterio de ordenación.

5.1. Diferencias entre Reentrenamiento y coentrenamiento

Reentrenamiento es en realidad un conjunto de coentrenamientos si atendemos a que utilizamos dos clasificadores binarios por cada uno de los sentidos de la palabra. No obstante, hay diferencias en estos aprendiza-

jes parciales que los alejan de la propuesta inicial de coentrenamiento.

En el coentrenamiento definido por (Blum y Mitchell, 1998) se mantiene la proporción entre positivos y negativos, respetando las frecuencias de la propia semilla. Reentrenamiento no tiene en cuenta esta característica, incorpora todos los ejemplos que obtienen la máxima probabilidad si no hay colisiones entre clases (propuesta común).

Lo cierto es que un ejemplo que no sea elegido en una iteración por este límite, por lo general, acaba siendo elegido en las siguientes. De hecho, el aspecto de la proporción de negativos y positivos no se ha tenido en cuenta dado que, por el filtro de la propuesta combinada, ciertos sentidos tienen dificultades para ganar positivos y, simplemente, no se puede satisfacer la proporción. Así, el coentrenamiento está definido para un número finito de iteraciones, mientras que Reentrenamiento puede detenerse antes de la última iteración porque no es capaz de clasificar ningún contexto con la suficiente seguridad.

Tampoco se procesan los negativos como tales, simplemente se eligen de entre los positivos de otras clases, lo que puede dar lugar a menos negativos de los esperados (por ejemplo, que sólo se incorpore un positivo a la siguiente iteración y que una clase espere dos o más negativos). En el coentrenamiento se eligen tanto positivos como negativos en la proporción definida. En Reentrenamiento, la elección de negativos es más complicada debido a la concurrencia de varias propuestas de positivos y negativos en la misma iteración.

El coentrenamiento parte de una semilla cuyos miembros son elegidos aleatoriamente, mientras nosotros utilizamos todo el CA. En las pruebas que realizamos, previas al estudio que aquí se va a desarrollar, la elección aleatoria nos daba resultados muy dispares en diferentes ejecuciones, por lo que la descartamos.

La única razón por la que se nos ocurre que un ejemplo no deba ser parte de la semilla tiene que ver con el aprendizaje activo y es que ese ejemplo en particular sea dañino por sí mismo, que no sea adecuado para el entrenamiento. No hemos realizado estudio alguno respecto a este asunto.

Otro aspecto no suficientemente aclarado por (Blum y Mitchell, 1998), ya que dan a entender que sus dos clasificadores podrían trabajar en paralelo, es qué ocurre cuando am-

Los clasificadores proponen el mismo ejemplo como positivo, aspecto éste que está asumido en Reentrenamiento.

6. Evaluación de la anotación por Reentrenamiento

Este experimento tiene por objeto comprobar sobre un corpus anotado si Reentrenamiento acierta al clasificar. Se han utilizado dos corpus distintos, el *Interest Corpus* (Bruce y Wiebe, 1994) en primer lugar, por su extensión, y el DSO en segundo, por cubrir más palabras y tener una extensión media. En todo caso, las palabras escogidas son nombres.

Tras la evaluación con el corpus Interest, se demostrará empíricamente, la validez del método, concretamente la ganancia frente a estrategias más cercanas al coentrenamiento, y la comparación con los resultados que se obtendrían con esquemas de desambiguación más simples, todo ello, ahora, con el corpus DSO.

6.1. Pruebas con el *interest corpus*

En esta primera prueba, se ha dividido el corpus Interest en 5 partes, lo que permite definir 5 parejas de CA y CNA (1 parte como CA y 4 como CNA), correspondientes cada una a un reentrenamiento que comienza con una semilla diferente. En todos los casos los atributos han sido (*LB-SDM*).

El Cuadro 1 muestra los resultados promedio de estas pruebas en precisión, cobertura, F1, y cobertura absoluta, respectivamente. Para cada configuración de umbral, se muestran los valores correspondientes a las primeras 100 iteraciones (primer fila) y los datos obtenidos al final del proceso (segunda fila de cada umbral).

Como primer hecho relevante, Reentrenamiento tiene un buen comportamiento tanto en precisión como en cobertura. Todas las pruebas han conseguido una precisión por encima del 89%. El umbral 0,8 – 0,4 obtiene las mejores precisiones pero la cobertura absoluta es, evidentemente, también la más baja. La relajación de este parámetro se traduce en más ejemplos y una disminución gradual de la precisión al ir reduciendo el umbral exigido. En general, los resultados finales se han conseguido al detenerse el proceso entre las iteraciones 1000 y 1300, dependiendo precisamente de lo alto que es el umbral.

Atributos: LB-SDM				
Umbral	Pre	Rec	F1	Cob
0.0-0.0	94,2	10,4	18,7	11,0
	92,4	67,7	78,1	73,3
0.1-0.1	97,1	9,0	16,4	9,2
	92,4	66,3	77,2	71,8
0.2-0.2	97,8	8,8	16,2	9,0
	89,6	67,5	77,0	75,3
0.3-0.3	98,5	8,1	15,0	8,2
	89,4	67,5	76,9	75,5
0.4-0.4	98,8	6,9	13,0	7,0
	89,2	70,7	78,9	79,3
0.5-0.4	99,0	6,5	12,2	6,6
	90,4	68,3	77,8	75,5
0.6-0.4	99,5	6,6	12,4	6,7
	94,0	64,3	76,4	68,3
0.7-0.4	99,8	6,0	11,3	6,0
	98,0	59,6	74,1	60,8
0.8-0.4	100,0	5,9	11,2	5,9
	98,9	49,2	65,7	49,7

Cuadro 1: Corpus Interest: resultados promedio con las cinco semillas en las 100 iteraciones iniciales y en la última iteración

La diferencia favorable al umbral 0,0 – 0,0 frente a los umbrales menores de 0,5 – 0,4 muestra que la disminución dinámica de este parámetro es excesiva si de mantener la precisión se trata, aunque permite incrementar la cantidad de ejemplos obtenidos.

En cuanto a la cobertura de sentidos, de los 6 posibles sentidos que aparecen en el corpus, Reentrenamiento consigue ejemplos de la mayoría de ellos, de 4 en la ejecución con más restricciones (0.8-0.4), y de todos con el umbral 0.0-0.0 (véase el Cuadro 2).

Umbral	Sentidos
0.0-0.0	6,0
0.1-0.1	6,0
0.2-0.2	5,5
0.3-0.3	5,2
0.4-0.4	5,3
0.5-0.4	5,8
0.6-0.4	5,1
0.7-0.4	4,5
0.8-0.4	3,9

Cuadro 2: Corpus Interest: promedio de cobertura de sentidos (de 6 posibles)

También es cierto que el aprendizaje con cada una de las semillas obtiene buenos resultados sin reentrenamiento, como se puede ver en el Cuadro 3 que se muestra como referencia. En este cuadro pueden verse los da-

tos de una clasificación “normal”: se aprende con la semilla y se clasifica el CNA. Comparadas con las de Reentrenamiento, se observa en éstas una ganancia significativa en precisión. Los datos de precisión y F1 se muestran de forma gráfica en las Figuras 3 y 4.

Umbral	Atributos	Pre	Rec	F1	Cob
0,0	LBSDM	81,1	80,8	81,0	99,6
0,0	LB	83,7	70,9	76,8	84,7
0,0	SDM	80,2	79,9	80,1	99,5
0,5	LBSDM	90,7	63,6	74,8	70,1
0,5	LB	93,4	58,7	72,1	62,8
0,5	SDM	91,0	63,2	74,6	69,4
0,8	LBSDM	90,0	54,8	68,1	60,8
0,8	LB	93,3	50,0	65,1	53,6
0,8	SDM	90,7	55,1	68,6	60,8

Cuadro 3: Corpus Interest: valores de referencia con el sistema ME-WSD

Las pruebas realizadas para cada umbral consisten en el aprendizaje y clasificación utilizando sólo uno de los grupos de atributos que definen las dos vistas utilizadas en este experimento, y también con la conjunción de los dos.

Los datos de precisión en los umbrales extremos (0.8 y 0.0) muestran una ganancia clara entre Reentrenamiento y ME-WSD, al tiempo que los valores de F1 están muy cercanos. En los umbrales intermedios (0.5 es el que se muestra en este artículo) las diferencias no son tan claras dependiendo de lo estricto que sea este parámetro, aunque no hace más que confirmar que Reentrenamiento mantiene ese buen comportamiento puesto que, en el peor caso, se puede considerar que los resultados son parejos entre uno y otro proceso, y no dependen tanto de una correcta elección de atributos.

6.1.1. Selección global secuencial

El siguiente experimento está más enfocado a la aplicación en competiciones tipo SENSEVAL, es decir, queremos comprobar si un esquema de reentrenamiento aportaría ventajas frente a un proceso normal de aprendizaje-clasificación.

La hipótesis es que cada grupo de atributos es capaz de detectar la clase de un pequeño conjunto de contextos y, además, con una alta precisión, ayudado por los datos obtenidos del 3FCV sobre el corpus de entrenamiento que permiten una definición adecuada de los conjuntos de atributos que definen las dos vistas de los datos.

PARCIAL		ACUMULADO					
t	pre	t	pre	cob	rec	f1	
337	92,3	337	92,3	15,1	14,0	24,3	R1
389	77,6	726	84,4	32,6	27,6	41,5	R2
264	65,2	990	79,3	44,5	35,3	48,8	R3
21	76,2	1011	79,2	45,4	36,0	49,5	R4
1214	56,9	2225	67,2	100,0	67,1	67,1	F

ATRIBUTOS	
at1	at2
R1: C	–
R2: W	–
R3: Lbc	IBC
R4: SWr	LCd
F: sk5(*)	

*clasificación estándar, sin reentrenamiento

Cuadro 4: Evaluación Reentrenamiento secuencial sobre corpus test SENSEVAL-2 español: precisión, cobertura y F1

Siendo el corpus de entrenamiento el CA y el de test el CNA, se plantean cuatro reentrenamientos sucesivos de tal forma que la salida de uno es la entrada del otro; el corpus de aprendizaje del siguiente reentrenamiento está engrosado con los contextos clasificados por el anterior, al mismo tiempo que el CNA va disminuyendo de tamaño. Finalmente, se entrena un clasificador ME con el corpus de aprendizaje resultado de los anteriores y se clasifican el resto de instancias no cubiertas.

En el Cuadro 4 se muestran los resultados del experimento. Cada reentrenamiento viene identificado por Rx y la clasificación final por F , y se acompaña de los conjuntos de atributos utilizados en cada ejecución. Este cuadro detalla los valores parciales obtenidos en cada fase de reentrenamiento (*PARCIAL*), presentando los aciertos (a), errores (e), la suma de ambos (t) y, finalmente, la precisión alcanzada (pre); los valores acumulados (*ACUMULADO*) muestran, además, la cobertura absoluta (cob) y la cobertura (rec , de *recall*).

Los dos primeros reentrenamientos, $R1$ y $R2$, se configuran con un único clasificador parcial, aprovechando la alta precisión del grupo de atributos, en nuestro experimento C y W , respectivamente. No se ha utilizado el B por ser muy similar al C . Los otros dos reentrenamientos, que se supone deben clasificar contextos de una mayor dificultad, se configuran con dos selecciones de grupos de atributos atendiendo a su precisión y al tipo

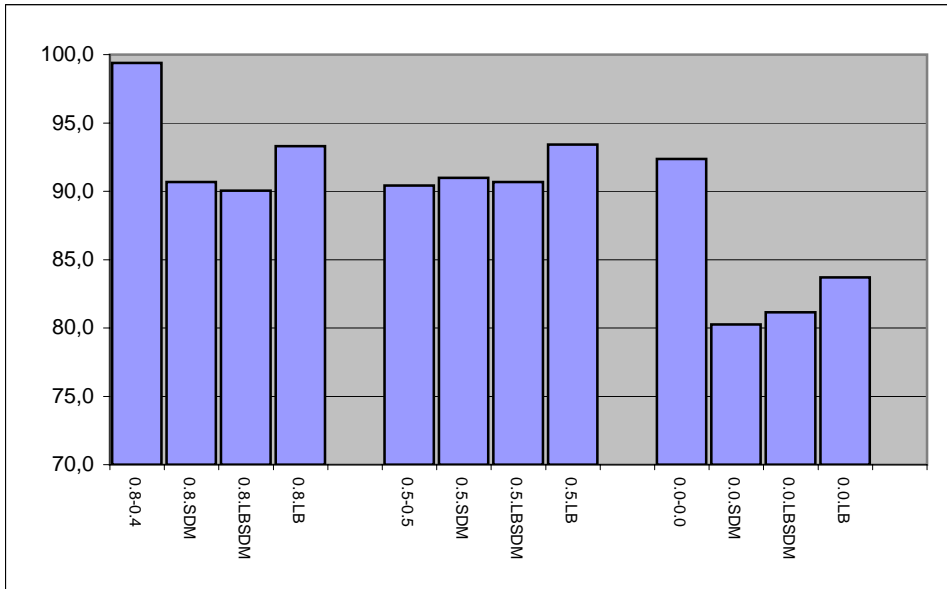


Figura 3: Comparación de precisiones entre Reentrenamiento y ME-WSD

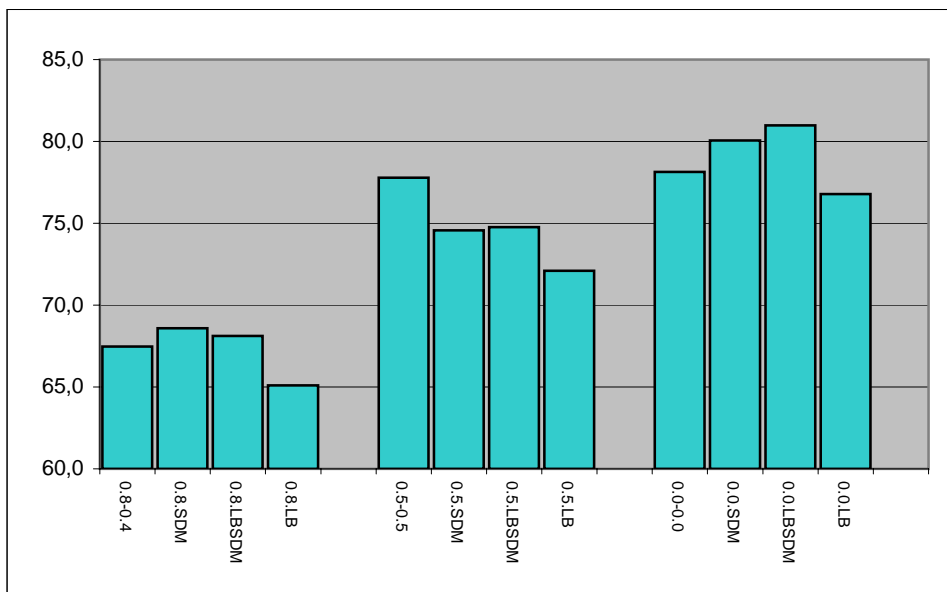


Figura 4: Comparación de F1 entre Reentrenamiento y ME-WSD

de información que procesa. Concretamente, el tercer reentrenamiento se basa en las mínimas diferencias entre atributos relajados y no relajados (*Lbc - lBC*), buscando nuevamente los atributos con una buena precisión al tiempo que asegurar una cierta cobertura absoluta. El cuarto ya presenta diferencias notables en el tipo de información tratada (*SWr - LCd*).

La clasificación final pretende cubrir ya el total de contextos todavía en el CNA, apren-

diendo con el corpus generado por el cuarto reentrenamiento.

Efectivamente, el primer reentrenamiento (sólo el grupo *C*) consigue un 92% de precisión, aunque con una cobertura muy baja del 15%. Sin embargo, el segundo reentrenamiento (*W*) consigue clasificar casi la misma cantidad de contextos, siendo en este caso la precisión del 77%. El tercer reentrenamiento ya utiliza clasificadores parciales diferenciados (*Lbc - lBC*) pero obtiene tan sólo un

65 % de precisión y clasifica menos contextos. El último reentrenamiento se configura de manera aún más restrictiva (*SWr - Lcd*) alcanzando el 76 % de precisión pero clasificando sólo 21 instancias nuevas.

En todos los experimentos realizados hasta ahora sobre Reentrenamiento se ha visto un decremento de la precisión al aumentar la cobertura, hecho éste que se repite aquí. El objetivo era clasificar tantos contextos como pudiéramos pero asegurando una alta precisión.

Observemos, ahora, los valores acumulados en el mismo Cuadro 4. Al llegar al cuarto reentrenamiento, la menor precisión de las últimas ejecuciones se compensa precisamente con esa menor cobertura, y la precisión y cobertura acumuladas, es decir, teniendo en cuenta las clasificaciones efectuadas hasta entonces, son del 79 % (precisión) y 45 % (cobertura).

El CNA restante es considerado como difícil de clasificar y, efectivamente, aplicado un entrenamiento normal la precisión acumulada decae hasta el 67 %.

Podemos comparar estos valores con los que se obtienen de la evaluación de un clasificador entrenado con *sk5* a partir de únicamente el corpus de aprendizaje, como se puede consultar en el Cuadro 5

	sk5	reent	dif
Todos	62,5	67,1	+4,58
Nombres	61,2	67,0	+5,76
Verbos	52,1	57,2	+5,10
Adjetivos	75,3	78,2	+2,64

Cuadro 5: Comparación de un clasificador entrenado con *sk5* y los reentrenamientos secuenciales: tasas de acierto

El Reentrenamiento secuencial supone una mejora sobre el clasificador *sk5* de casi un 5 %. Todas las categorías ven incrementada su tasa de acierto, siendo los adjetivos en los que se observa una diferencia menor. Es obvio, pues, que el efectuar varios reentrenamientos previos a la clasificación normal aumenta la cantidad de aciertos finales.

Sin embargo, el hecho remarcable es que somos capaces de asegurar una precisión cercana al 80 % con reentrenamientos sucesivos, a falta de concretar qué hacer con estas clasificaciones. Puede que determinadas tareas del PLN precisen de una determinada preci-

sión aún cuando la cobertura absoluta no sea total. También es plausible la posibilidad de que la última fase de clasificación se haga con otros métodos (si es que se pretende clasificar el 100 % de los contextos de test, que con el Reentrenamiento es difícil de conseguir).

En el Cuadro 6 se puede ver el detalle del Reentrenamiento secuencial por categorías.

TODOS		NOMS.		VERS.		ADJS.		
pre	rec	pre	rec	pre	rec	pre	rec	
92,3	14,0	90,2	13,8	88,2	11,0	97,5	17,5	R1
84,4	27,6	85,0	24,2	74,8	21,1	91,0	38,6	R2
79,3	35,3	79,6	31,2	68,3	27,8	88,0	48,3	R3
79,2	36,0	79,3	32,0	68,0	27,9	88,2	49,5	R4
67,1	67,1	67,0	67,0	57,2	57,2	78,0	78,0	F

Cuadro 6: Evaluación Reentrenamiento secuencial sobre corpus test SENSEVAL-2 español: detalle por categorías

Los adjetivos y los nombres son los que mantienen unos valores de precisión altos durante los reentrenamientos, pero también se destacan los verbos por conseguir precisiones por encima del 68 %. En todas las categorías, la última clasificación, la que debe procesar los contextos que aún no han podido ser clasificados, es la que hace caer la precisión. Cómo se ha hecho anteriormente, podríamos diferenciar los reentrenamientos por categorías, buscando esa mejora final en los verbos.

El porqué realizar esa clasificación final se encuentra en nuestro deseo de llegar a cubrir el 100 % de los contextos de test. No obstante, los reentrenamientos anteriores han dificultado la tarea de este último clasificador puesto que ya han detectado y procesado los contextos más fáciles de clasificar.

6.2. Senseval-3

El Reentrenamiento fue probado en el último SENSEVAL-3 (*Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*) en las tareas de muestra léxica en inglés y en español. En la primera, el objetivo era comprobar que el sistema era capaz de asegurar una precisión suficientemente alta. En la segunda, se utilizó como apoyo al sistema “normal” basado en ME, aplicándolo a las palabras que, a priori, serían más difíciles de clasificar basándonos en la distribución de sentidos en el corpus de aprendizaje.

Los resultados en este SENSEVAL no fueron todo lo buenos que esperábamos. Como

se puede comprobar en el Cuadro 7 (los sistemas de la izquierda son los presentados por nosotros, y los de la derecha los que obtuvieron los mejores resultados en cada tarea) la precisión obtenida para la tarea en inglés, aun siendo notable, no supone una gran ventaja sobre el mejor sistema que consiguió una cobertura absoluta total mientras que el nuestro tan sólo cubrió una tercera parte de las clasificaciones posibles.

El resultado en la tarea para el español tampoco fue satisfactoria ya que, en este caso, el objetivo era la respuesta al 100 % de las instancias de test y Reentrenamiento sólo se aplicó a un subconjunto de las palabras objetivo. Se pretendía ayudar a mejorar la precisión de las palabras para las que la distribución de sentidos era más equilibrada, lo que, en principio, puede hacer prever una menor eficacia. Sin embargo, el reentrenamiento influyó negativamente, y como se verá más adelante, el sistema "normal" hubiera obtenido algo menos de dos puntos más de precisión.

Tarea	Pre	Rec	Sistema	Pre	Rec
español	84,0	84,0	IRST	84,2	84,2
inglés <i>fine</i>	78,2	31,0	htsa3	72,9	72,9
inglés <i>coarse</i>	82,8	32,9	IRST-Kernels	79,5	79,5

Cuadro 7: Algunos resultados de los sistemas supervisados de la Universidad de Alicante en SENSEVAL-3

A continuación, pasamos a describir en detalle el sistema presentado para la tarea en español en SENSEVAL-3 y, también, otros experimentos realizados con posterioridad con los mismos datos.

6.2.1. La tarea en español

Los datos para la tarea en español estaban distribuidos en tres conjuntos: un conjunto anotado con sentidos para entrenamiento (CA), otro no anotado para test (T), y un tercero sin anotar (CNA), bastante más amplio que el anotado, que los investigadores podían utilizar como creyeran conveniente⁴. Nuestra intención era aumentar los conjuntos de aprendizaje de estas palabras aplicando Reentrenamiento, con la esperanza de que una mayor cantidad de ejemplos ayudara a mejorar la eficacia del sistema globalmente.

⁴Al final, sólo nosotros hicimos uso de esta tercera parte.

Las palabras que se seleccionaron (24 de 46) cumplían que el sentido más frecuente no suponía más allá de un 70 % del total de ejemplos de entrenamiento en el CA. Los ejemplos de estas palabras constituyeron la semilla para Reentrenamiento. La definición de las dos vistas de atributos a usar en cada pareja de clasificadores se hizo mediante un estudio de los grupos de atributos más precisos para cada palabra, concretamente los seis más precisos se alternaron en cada vista.

Así, ciertas palabras fueron entrenadas con los corpus CA sin modificar, y el resto con los corpus aumentados mediante Reentrenamiento. Concretamente para éstas últimas, se pudo comprobar que los nuevos ejemplos, muy probablemente, provocaron un sobreentrenamiento que se cuantifica en una ligera disminución global de la eficacia de los clasificadores, tal y como puede verse en el Cuadro 9 (sistema *Sval3*): globalmente, se clasificaron mal 66 instancias del conjunto T, comparándolo con la ejecución sin reentrenamiento previo (sistema *normal*), y el sistema sufrió una pérdida de precisión de un 1,6 %.

Una de las causas que, pensamos, provocaron este comportamiento no deseado es que Reentrenamiento no fue diseñado para respetar la distribución de sentidos en el corpus semilla. En el Cuadro 8 muestra el resumen del incremento de los corpus de entrenamiento. Se consiguieron 2006 nuevos ejemplos para el subconjunto de palabras seleccionado (un 46 % más de ejemplos), con una tendencia general a generar clasificaciones de los sentidos más frecuentes en detrimento de los otros (un 59 % más de ejemplos si se tiene en cuenta únicamente el sentido más frecuente).

	Todos SMF	
Original	4390	2207
Nuevo	6396	3542
Incremento	1,46	1,59

Cuadro 8: Aumento de ejemplos en Senseval-3 español

6.2.2. Nuevos experimentos

Una vez que ya disponíamos de la anotación correcta de test, después de celebrado SENSEVAL-3, preparamos nuevos experimentos para comprobar si otras formas de aplicar Reentrenamiento podrían mejorar los resultados obtenidos en la competición.

Se ha probado el Reentrenamiento secuen-

cial directamente sobre los corpus de entrenamiento y test de la muestra léxica en español de SENSEVAL-3. Concretamente realizamos dos experimentos con diferentes configuraciones de atributos, una basada en el estudio por 3FCV sobre el conjunto de aprendizaje de la precisión de los atributos, y otra que es la misma que la del Cuadro 4. Los resultados se muestran en el Cuadro 9, donde se puede ver la precisión de cada uno de las ejecuciones: el sistema ME-WSD sin reentrenamiento (“Normal”), el sistema que compitió en SENSEVAL-3 (“Sval3”), y los dos nuevos sistemas mencionados (“E1” y “E2” respectivamente).

sistema	precisión	(*)Atributos	
		at1	at2
Normal	85,6		
E1(*)	85,3	R1: c	–
Sval3	84,0	R2: lW	oj
E2	82,1	R3: sp	oi
		R4: LPi	OSj
		F:	0lWsbcpij
		Umbral: 0.6-0.3	

Cuadro 9: Comparación de sistemas para Reentrenamiento secuencial

En primer lugar, vuelve a ponerse de manifiesto que la selección de atributos es necesaria, no todas las palabras, ni la mismas palabras pero en distintos corpus, se pueden aprender con los mismos atributos. El sistema *E2* estaba configurado con los atributos del Cuadro 4, que se calcularon para los datos del SENSEVAL-2, y el sistema *E1* utilizaba los datos de precisión de los grupos de atributos del 3FCV sobre el corpus de SENSEVAL-3.

Además, y dado que tal información estaba disponible en la tarea de español de SENSEVAL-3, se definieron 2 grupos nuevos de atributos (*i* y *j*) basados en los códigos IPTC y ANPA que completaban la anotación semántica de los datos de aprendizaje y test. Esta información se utilizó tanto en la propia competición como en los experimentos que estamos describiendo ahora mismo.

En el Cuadro 10 se muestra el resumen de la evaluación del sistema *E1*.

Las fases de reentrenamiento (de la 1 a la 4) consiguen precisiones muy altas, sobre todo las dos primeras. La fase 1 se vale de una única vista que son las composiciones de palabras en el entorno cercano de la palabra objetivo (atributos *c*) dado que el análisis sobre el corpus de entrenamiento le fue muy favo-

Parciales	1	2	3	4	F
precisión	96,7	96,7	92,3	82,9	62,6
cob. abs.	26,3	16,8	24,5	3,5	28,9
Acumulados					
precisión	96,7	96,7	95,1	94,5	85,3
cobertura	25,4	41,7	64,3	67,2	85,3
F1	40,3	58,3	76,7	78,5	85,3
cob. abs.	26,3	43,1	67,6	71,1	100,0

Cuadro 10: Datos de evaluación del sistema *E1*

nable. Las siguientes fases combinan grupos de atributos, de más precisos a menos, buscando la complementariedad de las clasificaciones, esto es, que cada una detecte aquellas anotaciones que le son más evidentes y que no necesariamente han de clasificar las otras fases. Si observamos la parte del cuadro que muestra los resultados parciales de precisión y cobertura absoluta, es cierto que la fase 4 muestra síntomas de agotamiento del método puesto que ya se había clasificado casi las tres cuartas partes del corpus de test (sólo clasificó un 3,5 % del total de ejemplos).

Por otro lado, viendo los datos acumulados, al llegar a la fase 4 la degradación de la precisión es mínima si la comparamos con la obtenida en la primera (de 96,7 % a 94,5 %). Es decir, clasificando un 71 % de los ejemplos de test, conseguimos una precisión altísima para lo que es la tarea de WSD en SENSEVAL.

Sin embargo, la última fase (F) que realiza un entrenamiento-clasificación estándar, aunque partiendo del corpus de aprendizaje conseguido en la fase 4, la precisión decayó de forma muy importante por una razón fundamental y ya comentada: los ejemplos que aún quedaban por clasificar son, precisamente, los que más dificultades presentan a la hora de etiquetarlos. Seguramente, se podría forzar la fase 4 (bajando el umbral, por ejemplo) para que anotara la mayor cantidad posible del corpus restante de test.

Otro aspecto importante es la variación de la distribución de sentidos al terminar la fase 4. Puesto que, en este punto, las instancias de test que faltan por clasificar son, aparte de las de menor confianza para método, se ven perjudicadas por su menor peso estadístico en el modelo. De hecho, se consigue una mejora de 2 décimas si se establece un umbral nulo en la fase 4 y la fase F se entrena con el corpus original y no con el obtenido en las etapas de

reentrenamiento. Esto no significa necesariamente una contradicción con el experimento con los datos de SENSEVAL-2 (véase el Cuadro 4), ya que la última competición supuso un salto cualitativo importante, creemos que fruto de la confección de los datos de entrenamiento y test.

Finalmente, es una prueba más de que la anotación conseguida por Reentrenamiento es fiable, aún cuando su aplicación a competiciones de tipo SENSEVAL no está clara, ya que estamos casi seguros de que los corpus incrementados en la competición (sistema *Sval3* en el Cuadro 9) están fundamentalmente bien anotados. Sin embargo, cuando comparamos los resultados contra los que se obtienen con los corpus originales, aún no siendo un resultado totalmente rechazable, sí es decepcionante (aunque aún falta por probar qué hubiera pasado si el reentrenamiento y los corpus obtenidos desde el corpus no anotado se hubieran aplicado a todas las palabras).

7. Conclusiones

Uno de los retos de la WSD supervisada actualmente es la necesidad de adquirir grandes cantidades de datos anotados para elevar su eficacia y hacerla útil para otras tareas del PLN. La disponibilidad de tales recursos, especialmente para los idiomas que no son el inglés, es insuficiente y demasiado costosa (en términos de tiempo y personal) pero, sin embargo, la cantidad de texto sin anotar y al alcance de todos es inmensa.

En este artículo, utilizamos texto no anotado con un algoritmo iterativo-incremental que adquiere automáticamente ejemplos anotados con una alta precisión. El núcleo del proceso es, a su vez, un sistema supervisado de WSD basado en los modelos de máxima entropía que ha probado su competitividad, aunque es evidente que cualquier método de clasificación supervisado es aplicable. A este algoritmo iterativo lo hemos denominado *Reentrenamiento* ya que es una aplicación multiclase del algoritmo de coentrenamiento (*co-training*) propuesto por (Blum y Mitchell, 1998), al que se le añaden, además, condiciones más estrictas a la hora de decidir si un determinado ejemplo puede etiquetarse o no.

El *Reentrenamiento* consigue precisiones por encima del 90 % en el corpus Interest y en la tarea para español de SENSEVAL, al tiempo que la cobertura absoluta y de sentidos es buena. Al contrario que *coentrenamiento*, es

más difícil que se produzca una degradación excesiva de la precisión ya que el proceso se detiene cuando ninguno de los ejemplos candidatos para la siguiente iteración ofrece suficientes garantías de estar bien clasificados.

Este método es complementario con otros métodos de adquisición semisupervisada de nuevos conjuntos de ejemplos etiquetados con sentidos, ayudando todos a disminuir el esfuerzo exigido a los anotadores y a superar el cuello de botella que supone la adquisición de conocimiento suficiente.

Como trabajo futuro pretendemos estudiar la naturaleza de las anotaciones, por qué ciertos sentidos (no siempre los más frecuentes en el corpus) tienen más “éxito” que otros, lo que provoca una alteración de la distribución con respecto del corpus inicial notable. Así mismo, el uso de otros métodos supervisados de aprendizaje automático que requieren un menor coste computacional, incluso la combinación de varios de ellos y no solamente dos, nos proporcionará la base para acometer proyectos de anotación ambiciosos.

Bibliografía

- Abney, Steven. 2002. Bootstrapping. En *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, páginas 360–367.
- Abney, Steven. 2004. Understanding the Yarowsky Algorithm. *Computational Linguistics*, 30(3):365–395.
- Blum, Avrim y Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. En *Proceedings of the 11th Annual Conference on Computational Learning Theory*, páginas 92–100, Madison, Wisconsin, July. ACM Press.
- Bruce, Rebecca y Janyce Wiebe. 1994. Word sense disambiguation using decomposable models. En *Proceedings of the ACL-94, 32nd Annual Meeting of the Association for Computational Linguistics*, páginas 139–145, Las Cruces, US.
- Collins, Michael y Yoram Singer. 1999. Unsupervised models for named entity classification. En Pascale Fung y Joe Zhou, editores, *Proceedings of 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, páginas 100–110, Maryland, USA. ACL.

- Edmonds, Phil. 2000. Designing a task for senseval-2. Informe técnico, Senseval-2 website.
- Escudero, G., L. Màrquez, y G. Rigau. 2000. Boosting Applied to Word Sense Disambiguation. En *Proceedings of the 11th European Conference on Machine Learning, ECML-2000*, Barcelona, Spain.
- Manning, Christopher D. y Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.
- Mihalcea, Rada. 2003. Unsupervised natural language disambiguation: the role of non-ambiguous words. En *Proceedings of the RANLP'03*.
- Mihalcea, Rada. 2004. Co-training and self-training for word sense disambiguation. En *Proceedings of CoNLL-2004*, páginas 33–40. Boston, MA, USA.
- Ng, Hwee Tou y Hiang Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. En *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, páginas 40–47, University of California, Santa Cruz, CA.
- Nigam, Kamal y Rayid Ghani. 2000a. Analyzing the effectiveness and applicability of co-training. En *Proceedings of the 9th International Conference on Information and Knowledge Management*, páginas 86–93. ACM Press.
- Nigam, Kamal y Rayid Ghani. 2000b. Understanding the behavior of co-training. En *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Text Mining*, páginas 105–106.
- Pedersen, Ted. 2001. A decision tree of bigrams is an accurate predictor of word sense. En *Proceedings of the Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, páginas 79–86, Pittsburgh, July.
- Philips, Williams y Ellen Riloff. 2002. Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. En *Proceedings of Conference on Empirical Methods in Natural Language Processing EMNLP'02*.
- Ratnaparkhi, Adwait. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. tesis, University of Pennsylvania.
- Sarkar, Anoop. 2001. Applying co-training methods to statistical parsing. En *Proceedings of the 2nd Annual Meeting of the NAACL*, páginas 95–102, Pittsburgh, PA.
- Steedman, Mark, Rebecca Hwa, Stephen Clark, Miles Osborne, Anoop Sarkar, Julia Hockenmaier, Paul Ruhlen, Steven Baker, y Jeremiah Crim. 2003. Example selection for bootstrapping statistical parsers. En Marti Hearst y Mari Ostendorf, editores, *HLT-NAACL 2003: Main Proceedings*, páginas 236–243, Edmonton, Alberta, Canada, May 27 - June 1. Association for Computational Linguistics.
- Suárez, Armando. 2004. *Resolución de la ambigüedad semántica de las palabras mediante modelos de probabilidad de máxima entropía*. Ph.D. tesis, Universidad de Alicante, junio.
- Suárez, Armando y Manuel Palomar. 2002. A maximum entropy-based word sense disambiguation system. En Hsin-Hsi Chen y Chin-Yew Lin, editores, *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002*, páginas 960–966, Taipei, Taiwan, August.
- Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. En *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, páginas 189–196.