Procesamiento del Lenguaje Natural, nº 31, septiembre de 2003 ISSN 1135-5948

SEPLN

Sociedad Española para el Procesamiento del Lenguaje Natural

Artículos

XIX Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural

SEPLN

Universidad de Alcalá, 10, 11 y 12 de septiembre de 2003

Instituto Cervantes

UNIVERSIDAD DE ALCALÁ

Excmo. Ayuntamiento de ALCALÁ de HENARES
Concejalía de Turismo

CIUDAD PATRIMONIO de la Humanidad
ALCALÁ de Henares

MINISTERIO DE CIENCIA Y TECNOLOGIA

Telefónica IBERIA

FUNDACION MAPFRE ESTUDIOS

Comité organizador

Presidente:

D. Jesús Antonio Cid Martínez
Director Académico del Instituto Cervantes

Coordinadora:

Dña. Isabel Bermejo Rubio
Responsable de la Oficina del Español en la Sociedad de la Información (OESI) del Instituto Cervantes

Vocales:

Dña. Eva Mª García García
Técnico de la OESI del Instituto Cervantes

Dña. Raquel Tapias Aparicio
Técnico de la OESI del Instituto Cervantes

D. John Michael Urresti Graña
Técnico de la OESI del Instituto Cervantes

Colaboradores:

Dña. Rosario Guijarro Huerta
Colaboradora de la OESI del Instituto Cervantes

Asesores:

Dña. Esmeralda de Luis Martínez
Jefe del Dpto. de Relaciones Exteriores e Institucionales del Instituto Cervantes

Dña. Gloria Gamarra Alonso de Linaje
Técnico del Dpto. de Relaciones Exteriores e Institucionales del Instituto Cervantes

Procesamiento del Lenguaje Natural, nº 31, septiembre de 2003 ISSN 1135-5948

SEPLN

Sociedad Española para el Procesamiento del Lenguaje Natural

Artículos

Procesamiento del Lenguaje Natural, nº 31 (2003), pp-125-132 recibido 16-05-2003; aceptado 20-06-2003

Aprendizaje competitivo LVQ para la desambiguación léxica*

Manuel García Vega
Universidad de Jaén
Av. Madrid 31, 23071
mgarcia@ujaen.es

María Teresa Martín Valdivia
Universidad de Jaén
Av. Madrid 31, 23071
maite@ujaen.es

Luis Alfonso Ureña López
Universidad de Jaén
Av. Madrid 31, 23071
laurena@ujaen.es

Resumen: La resolución de la ambigüedad léxica mejora significativamente muchas tareas del procesamiento del lenguaje natural. Presentamos un desambiguador supervisado basado en el Modelo de Espacio Vectorial en el que sus pesos se entrenan con un algoritmo competitivo basado en el modelo de Kohonen, concretamente el LVQ. Para ello, hace uso de las distintas relaciones semánticas de WordNet y también del corpus SemCor. El desambiguador se evalúa haciendo una simulación de participación en la competición SENSEVAL-2. Como muestran los resultados, la posición obtenida es muy buena.

Palabras clave: WSD, LVQ, Redes neuronales, Aprendizaje competitivo, SENSEVAL, WordNet, SemCor

Abstract: Word Sense Disambiguation improves several tasks of Natural Language Processing. We present a supervised disambiguator based on Vector Space Model, where its weights are trained with a learning vector quantization argorithm based on the Kohonen Model (LVQ algorithm) and using different semantic relations of WordNet and SemCor corpus. We also include an evaluation making a simulation of participation in SENSEVAL-2, obtaining a good position.

Keywords: WSD, LVQ, Neural Nets, Competitive Learning, SENSEVAL, WordNet, SemCor

1 *Introducción*

El objetivo de este trabajo es la resolución de la ambigüedad léxica. Se trata de una importante tarea para cualquier sistema de procesamiento de lenguaje natural. La resolución de la ambigüedad léxica se presenta en aquellas palabras que en función del contexto pueden tener un sentido u otro.

La desambiguación (Word Sense Disambiguation, WSD) consiste en identificar el significado de una palabra en un determinado contexto dentro de un conjunto de candidatos determinado. La desambiguación no es un fin en sí misma, sino una tarea intermedia muy necesaria para algunas tareas del Procesamiento del Lenguaje Natural (PLN) (Wilks y Stevenson, 1998, Palomar et al. 2001) como Recuperación de Información (IR) (Gonzalo et al., 1998, Schütze, 1998), Traducción Automática (MT) (Brown et al., 1991), Categorización de Textos (TC) (Ureña, Buenaga y Gómez, 2001), Sistemas de Diálogo y Extracción de Información (IE) (Kilgarriff, 1997)...

En los últimos años se han propuesto varios enfoques de distinta naturaleza para WSD, que podemos clasificarlos de acuerdo con la fuente de conocimiento utilizada (Ide y Veronis, 1998). Algunos enfoques se basan en la utilización de algún tipo de base de datos léxica (Xiaobin y Spakowicz, 1995). Otros, hacen uso de corpus de texto anotados semánticamente como colección de entrenamiento (Bruce y Janyce, 1994). Sin embargo, como la creación manual de corpus anotados semánticamente es una tarea difícil, tediosa y extremadamente

* Este trabajo ha sido financiado por el MCYT mediante el proyecto FIT-150500-2003-412

ISSN 1135-5948

costosa (Kilgarriff y Palmer, 2000) se han empleado también corpus no-anotados (Schütze, 1998, Pedersen y Bruce, 1997). Finalmente, recientes trabajos proponen la combinación de varias fuentes de conocimiento léxico (estructurado y no estructurado) así como diferentes técnicas y heurísticas para explotar dicho conocimiento (Wilks y Stevenson, 1998; Mihalcea y Moldovan, 2000, Ureña, Buenaga y Gómez, 2001).

En WSD, las técnicas aplicadas incluyen la utilización de métodos de aprendizaje automático. Los métodos de aprendizaje aplicados hasta el momento van desde los algoritmos de aprendizaje simbólico clásicos (árboles de decisión, inducción de reglas...) a los métodos subsimbólicos (redes neuronales, algoritmos genéticos...) pasando por los métodos de aprendizaje estocástico y por el aprendizaje no supervisado (como por ejemplo las técnicas de clustering).

En este trabajo se propone el uso de redes neuronales artificiales (RNA) para resolver la ambigüedad léxica. Concretamente, se aplica el algoritmo de aprendizaje por cuantificación vectorial (LVQ - Learning Vector Quatization).

La gran ventaja de las RNA es su capacidad de aprender a partir de variables que identifican el problema, extrayendo los datos necesarios para generar un modelo y una red capaz de resolverlo y, sobre todo, partiendo de un conocimiento mínimo de la esencia del problema. De hecho, hay varios trabajos que demuestran que las RNA son al menos tan eficientes como otros algoritmos de aprendizaje en muchos problemas (Atlas et al., 1989; Shavlik, Mooney y Towell, 1991)

El resto del artículo se organiza como sigue: En primer lugar, se presenta una breve descripción de los trabajos más relevantes en WSD usando RNA. La siguiente sección, se describe el algoritmo LVQ utilizado en la implementación de nuestro desambiguador. A continuación, se muestran los experimentos realizados así como los resultados obtenidos. Por último, se presentan las conclusiones y los trabajos futuros.

2 *Redes neuronales aplicadas a WSD*

Las RNA fueron originalmente una simulación abstracta de los sistemas nerviosos biológicos, formados por un conjunto de unidades llamadas "neuronas" o "nodos" dispuestas en distintas capas y conectadas unas con otras mediante unos pesos de conexión que permiten almacenar la información. Estas conexiones tienen una gran semejanza con las "dendritas" y los "axones" en los sistemas nerviosos biológicos. Las RNA pueden adquirir, almacenar y utilizar conocimiento experimental, obtenido a partir de ejemplos. Para ello, hacen uso de un algoritmo de aprendizaje mediante el cual se ajustan los pesos de conexión entre neuronas.

El procesamiento de información en una RNA comprende, generalmente, dos fases: una fase de entrenamiento y una fase de producción. Durante el entrenamiento, la red ajusta los pesos de conexión entre las distintas unidades siguiendo un determinado algoritmo de aprendizaje. El objetivo del entrenamiento es encontrar un conjunto de pesos para los que la aplicación de un conjunto de vectores de entrada genere el conjunto de salidas deseadas. Una vez que la red se estabiliza, es decir, no hay modificación de los pesos, comienza la explotación de la red. Se aplican unos valores de entrada, se propaga la señal a través de la red y se obtienen los valores de salida.

Aunque la bibliografía sobre RNA es muy extensa, una buena introducción se puede encontrar en (Fiesler y Beale, 1997).

El interés generado por las RNA tiene su justificación en las fascinantes propiedades que poseen entre las que destacan la capacidad de adaptación y autoorganización, memoria distribuida, procesamiento paralelo y capacidad de generalización. Debido precisamente a estas características, las RNA tienen aplicación en una gran variedad de disciplinas como ingeniería, física, biología, medicina, industria, finanzas... De hecho, las RNA se han utilizado con éxito para resolver un gran número de problemas del mundo real, existiendo una amplia gama de aplicaciones comerciales disponibles para distintas áreas. Por ejemplo, han sido utilizadas con éxito en tareas de clasificación de patrones (Bishop, 1995), tratamiento de señales (Widrow y Stearns, 1985), optimización (Cichocki y Unbehauen. 1993)...

Si bien, la cantidad de trabajos relacionados con la aplicaciones de RNA en muchos campos de la inteligencia artificial es desorbitada, concretamente, en el campo del procesamiento del lenguaje natural (PLN) la investigación no está aún muy desarrollada. No obstante, en la última década parece que el interés por relacionar las dos disciplinas ha crecido de manera espectacular como se puede comprobar

en (Robert, Moisl y Somers, 2000). Sin duda, el tipo de red más utilizado para las aplicaciones PLN es la red de propagación hacia atrás (BPN – BackPropagation Network) descrita en (McClelland y Rumelhart 1986), aunque existen algunos intentos por utilizar otras arquitecturas. Algunos ejemplos del uso de RNA en tareas de PLN se presentan a continuación.

En (Martín, García y Ureña, 2003) se presenta un modelo neuronal basado en aprendizaje competitivo que permite la categorización de textos multilingües. Ruiz y Srinivasan (2002) utilizan un clasificador jerárquico de RNA en un sistema de recuperación de información. En (Kohonen et al., 2000) se describe un sistema basado en el algoritmo SOM (self-organizing map) para autoorganizar grandes cantidades de documentos teniendo en cuenta sus similitudes textuales. Un ejemplo de resolución de anáforas se puede encontrar en (Allen, 1987).

Concretamente, para WSD se han aplicado varios modelos de red aunque la mayoría son extensiones de la BPN que utilizan las neuronas distribuidas en varias capas (redes multicapa) y con conexiones tipo feedforward (las señales de las neuronas se propagan desde las capas inferiores hasta las capas superiores).

Los primeros trabajos fueron realizados por Cottrell y Small (1983). En Véronis e Ide (1990) se usan dos aproximaciones independientes para WSD, los diccionarios electrónicos y las RNA. Otro trabajo interesante, fue presentado por Towell y Voorhees (1998) que utilizan una red BPN para aprender el contexto de palabras muy ambiguas. Recientemente, Martín, García y Ureña (2002) han utilizado una red neuronal basado en el modelo de Kohonen que utiliza aprendizaje competitivo supervisado para WSD.

3 *Algoritmo LVQ*

En este trabajo se utiliza el algoritmo LVQ para WSD. El motivo de utilizar este algoritmo es que con él se han obtenido muy buenos resultados en otras tareas de PLN como categorización de texto (Martín, García y Ureña 2003), CLIR (Cross Lingual Information Retrieval) (Martínez et al., 2002) e, incluso, en WSD (Martín, García y Ureña 2002).

Este modelo neuronal LVQ utiliza un algoritmo de aprendizaje competitivo. En el aprendizaje competitivo sólo una unidad de salida está activa en cada momento. Todas las neuronas de una red competitiva reciben idéntica información de las unidades de entrada pero las unidades de salida compiten entre sí para ser la que se activa como respuesta a esa señal de entrada. Cada neurona se especializa en un área diferente del espacio de entradas y sus salidas se pueden utilizar para representar de alguna manera la estructura del espacio de entradas (autoorganización).

En 1982 Teuvo Kohonen presenta un modelo de red neuronal competitiva (Kohonen, 1995) con capacidad para formar mapas de características a través de una organización matricial de neuronas artificiales. El modelo presenta dos variantes denominadas SOM (Self Organizing Map) y LVQ (Learning Vector Quantization). La diferencia fundamental radica en el tipo de aprendizaje utilizado. Mientras que el algoritmo LVQ es supervisado, el modelo SOM no requiere el conocimiento de los datos de salida.

En cuanto a la arquitectura de red, el modelo LVQ utiliza una red de dos capas con N neuronas de entrada y M neuronas de salida. Cada una de las N neuronas de entrada se conecta a las M de salida mediante conexiones hacia delante (feedforward). Entre las neuronas de la capa de salida existen conexiones laterales, ya que cada una de ellas tiene influencia sobre sus vecinas a la hora de calcular los pesos de las conexiones hacia delante entre la capa de salida y la capa de entrada.

Para que la red LVQ aprenda un conjunto de patrones de entrada es necesario una fase de entrenamiento en la que se ajusten los pesos de conexión entre la capa de entrada y la capa de salida. Estos pesos se representan mediante un matriz W de NxM pesos. La red utiliza un aprendizaje competitivo de manera que las neuronas de la capa de salida compiten entre sí quedando una única unidad activa ante una determinada información de entrada a la red. Los pesos de conexión se ajustan en función de la neurona que haya resultado ganadora y únicamente se permite que aprenda a dicha la unidad.

La red LVQ utiliza aprendizaje supervisado por lo que requiere un conjunto de vectores de entrenamiento que describan el comportamiento de la red que se desea aprender $\{x_1,t_1\}$, $\{x_2,t_2\}...\{x_L,t_L\}$. Durante la fase de entrenamiento se presentan a la red este conjunto de informaciones de entrada (vectores

de entrenamiento) para que ésta establezca en función de la semejanza entre los datos las diferentes categorías (una por neurona de salida), que servirían durante la fase de funcionamiento para realizar la clasificación de nuevos datos que se presenten a la red. Los valores finales de los pesos de las conexiones entre cada neurona de la capa de salida con las de entrada se corresponderán con los valores de los componentes del vector de aprendizaje que consigue activar la neurona correspondiente.

El algoritmo de aprendizaje utilizado funciona de la siguiente manera (fórmula 1). Si x_i es clasificado correctamente, los pesos de la neurona ganadora, c, se hacen tender hacia x_i acercando el vector de w_c al vector x_i. Si por el contrario, x_i es clasificado incorrectamente, una unidad equivocada ganó la competición y por lo tanto sus pesos se deben alejar de x_i.

$$w_c = \begin{cases} w_c + \alpha(t)\cdot(x_i - w_c) \text{ si } c = d \\ w_c - \alpha(t)\cdot(x_i - w_c) \text{ si } c \neq d \end{cases} \quad (1)$$

donde $\alpha(t)$ es el ratio de aprendizaje, siendo $0<\alpha(t)<1$, una función monótona decreciente del tiempo. La inicialización de $\alpha(t)$ se hace con un valor cercano a cero, y decrece linealmente de manera que al final del proceso de aprendizaje su valor es prácticamente nulo (Kohonen, 1995).

4 *Descripción del desambiguador*

El desambiguador que se presenta está basado en el Modelo de Espacio Vectorial (MEV). Los sentidos de cada palabra son representados con vectores cuyas componentes son las distintas palabras de su contexto.

La bondad del desambiguador reside fundamentalmente en el peso que se le asigna a esas palabras y para ajustar estos valores usaremos la red LVQ. Los pesos de los vectores de entrada a la red se calculan según (Salton y McGill, 1983) con el estándar *tf·idf* y, tras el entrenamiento se obtienen los vectores prototipo con los pesos definitivos de cada sentido y para todas las palabras.

El vector que representa el contexto de cada palabra a desambiguar deberá compararse con cada uno de los vectores de sus sentidos, según la similitud del coseno:

$$sim(w_k, x_i) = \frac{w_k \cdot x_i}{|w_k| \cdot |x_i|} \quad (2)$$

El sentido representado por el vector de mayor similitud será el designado como sentido desambiguado (sentido ganador).

Para evaluar nuestro desambiguador, vamos a realizar los mismos experimentos de la competición SENSEVAL-2 English-Lex-Sample, pudiendo de esta forma comparar su comportamiento de manera robusta.

SENSEVAL (Kilgarriff, 1998) es una organización internacional dedicada a la evaluación de sistemas de desambiguación. Organiza la competición para comprobar la potencia y debilidad de los sistemas presentados. Las tareas propuestas para la competición son de diversa índole y en varios idiomas.

Para entrenar el desambiguador se han usado diversos recursos: SemCor 1.6, WordNet 1.6 y los corpus ofrecidos en la competición SENSEVAL-2.

Para la competición English-Lex-Sample, de SENSEVAL-2, la organización creó un corpus de entrenamiento para un total de 92 palabras, entre nombres, verbos y adjetivos altamente polisémicos, formado por 8.820 contextos y, por otro lado, un corpus de evaluación, que ofrecía un total de 4.255 contextos para esas mismas 92 palabras a desambiguar. Los textos de entrenamiento se han incorporado como entrada a la red LVQ mientras que el corpus de evaluación ha sido el que finalmente hemos usado para medir la bondad del desambiguador.

4.1 SemCor 1.6

En primer lugar, se ha usado el corpus SemCor 1.6 (Miller et al. 1993), que es el Brown Corpus etiquetado con los sentidos de WordNet 1.6, habiéndolo incluido en su totalidad: Brown-1, Brown-2 y Brown-v. En la Tabla 1 podemos ver un resumen estadístico sobre su composición.

Hemos usado el párrafo como unidad contextual y se han seleccionado todas las ocurrencias en SemCor 1.6 de las 92 palabras de la evaluación. De esta forma, se han preparado 2.495 contextos que son los pertenecientes a las palabras de evaluación.

Sin embargo, SENSEVAL-2 tiene etiquetados sus corpus con los sentidos de WordNet 1.7, por lo que hemos tenido que actualizar los sentidos[1] de SemCor 1.6, según Daudé J., Padró L. and Rigau G, 2001, para conseguir uniformidad en el proceso de evaluación.

[1] http://www.lsi.upc.es/~nlp/tools.html

	SemCor 1.6			
	BR1	BR2	BRV	Total
Palabras etiquetadas	198.796	160.936	316.814	676.546
Palabras con punteros a WN	106.639	86.000	41.497	234.136
Punteros a nombres	48.835	39.477	0	88.312
Punteros a verbos	26.686	21.804	41.525	90.015
Punteros a adjetivos	9.886	7.539	0	17.425
Punteros a adverbios	11.347	9.245	0	20.592
Sentidos distintos en nombres	11.399	9.546	0	20.945
Sentidos distintos en verbos	5.334	4.790	6.520	16.644
Sentidos distintos en adjetivos	1.754	1.463	0	3.217
Sentidos distintos en adverbios	1.455	1.377	0	2.832

Tabla 1: Datos de SemCor 1.6

4.2 WordNet

De WordNet 1.6 (Fellbaum, 1998) se han incorporado las relaciones semánticas de sinonimia, antonimia, hiperonimia, hiponimia, holonimia, meronimia y coordinados.

Como se muestra en Ureña, García y Martínez (2001) los sinónimos, antónimos, merónimos, etc. de una palabra dada pueden figurar en su contexto. La forma de incorporar esta información al entrenamiento ha consistido en crear párrafos con las palabras de cada relación. Así, por ejemplo, para el nombre "sense" se han generado 7 párrafos con los sinónimos de cada uno de sus 7 sentidos, 7 párrafos más con toda la jerarquía de hiperónimos correspondiente a cada sentido y así, para las relaciones consideradas.

Al igual que en el caso de SemCor 1.6, hemos actualizado los sentidos de las 92 palabras de la evaluación con el sentido adecuado en WordNet 1.7

4.3 SENSEVAL-2

Los corpus de la tarea English-Lex-Sample de SENSEVAL-2 (eng-lex-sample.training.xml y eng-lex-samp.evaluation.xml) son textos que están escritos en XML y no ofrecen ninguna información morfosintáctica, a excepción de la palabra a la que pertenece el contexto en cuestión, y está etiquetado con los sentidos de WordNet 1.7. Por tanto, sólo hemos tenido que generar los párrafos en el mismo formato que en los dos casos anteriores.

En la Figura 1, se muestra el formato común para todas las tablas generadas, ya sean de entrenamiento o de evaluación. Se observa que la palabra a la que pertenezca un contexto determinado, "art" y "authority" en nuestro ejemplo, está seguida por el carácter '\' y un número entero que simboliza su POS. A continuación figura un '#' y otro número que es el sentido de esa palabra en WordNet 1.7. Seguidamente aparece el número de palabras que hay en su contexto y, para terminar, se pueden observar todos los pares palabra-frecuencia de todas las palabras del contexto.

art\1#3 87 accused 1 all 1 amid 1 are 1 art 1 at 1 back 1 become 1 blunted 1 book 1 bounce 1 buchwald 1 but 1 debacle 1 ... sexual 1 sitting 1 slightly 1 soft 1 something 1 strict 1 surprising 1 survival 1 surviving 1 suzanne 1 takes 1 than 1 that 1 these 1 time 1 towns 1 unfolding 1 washington 1 weird 1 when 1 who 1 with 1 working 1 wright 1 writer 1 writing 1 yet 1
art\1#3 91 about 2 aggression 1 any 1 are 1 asserting 2 at 1 back 1 be 1 been 1 books 1 breakfast 1 but 3 can 2 cannot 1 capable 1 carry 1 ... talking 1 teach 1 that 3 therefore 1 trouble 1 turn 1 use 1 victim 1 want 1 way 2 what 1 whatever 1 who 1 will 2 win 1 with 1 wrong 1
art\1#4 95 almond 1 an 1 are 2 asked 1 away 1 be 1 black 1 boy 1 came 1 care 1 censored 1 colour 1 control 1 cover 1 crateload 1 create 1 cult 1 demurely 1 didnt 1 done 1 dressers 1 east 1 ... was 1 wed 1 wh 1 whatever 1 white 1 with 3 work 2 would 1 yet 1
authority\1#1 103 accidents 1 actions 1 activities 1 advice 1 all 1 amount 1 animals 1 any 1 as 3 badenpowells 1 be 2 become 1 boys 1 ... turn 1 valuable 1 view 1 was 3 were 2 when 1 which 3 with 4 woodcraft 1 workingclass 1

Figura 1: Formato de los ficheros de entrenamiento y evaluación de SENSEVAL

5 *Entrenamiento de la red LVQ*

Los vectores iniciales que necesita el algoritmo LVQ serán inicializados a cero, dejando actuar a la red neuronal de forma que en el entrenamiento de cada vector, introduzca los pesos adecuados dependiendo de si la elección del ganador ha sido o no correcta.

Se unen los tres corpus (SemCor, WordNet y Senseval) para formar un único corpus de entrenamiento.

Tenemos un total de 92 dominios de entrenamiento y un total de 13.065 vectores de entrenamiento y se deben procesar todos los dominios para calcular los pesos de los vectores según el MEV. Así pues, cada uno de los vectores de entrada será introducido en la red neuronal sucesivas veces. Conforme se va iterando, los pesos de los vectores prototipo se modifican en menor medida en función del ratio de aprendizaje $\alpha(t)$, comenzando con un valor

de 0,3 y decrementándose linealmente según la siguiente ecuación

$$\alpha(t+1)=\alpha(t)-\frac{\alpha(0)}{P} \quad (3)$$

donde P es el número total de pasadas del entrenamiento para un dominio determinado. Según (Kohonen et al., 1996), el número de pasadas P suele ser suficiente con 40 veces el número de vectores prototipos, sin embargo, debido al elevado número de vectores en cada dominio, se corre el riesgo de no entrenar algún sentido. En nuestro entrenamiento hemos introducido 25 veces cada vector, puesto que a partir de este numero de iteraciones la red se estabiliza.

Se trata de encontrar el sentido ganador calculando la menor distancia euclídea entre el vector de entrada, una vez pesado, y los vectores prototipo de la red LVQ. Una vez elegido el ganador, sus pesos serán ajustados utilizando la fórmula 1.

Para cada dominio, se emplea un tiempo en el orden de $O(25 \cdot f \cdot p \cdot \log_2 p)$, donde f es el número de ocurrencias de la palabra y p es el número promedio de palabras en sus ventanas. Al tratarse tan solo de 92 palabras a desambiguar, el proceso de entrenamiento ha llevado apenas unos minutos.

6 *Evaluación*

Del proceso de entrenamiento se obtiene un fichero con los 92 dominios con los pesos afinados, donde cada vector se corresponde con una palabra y sentido específico.

La evaluación es simple. Basta con calcular la similitud de cada vector de evaluación con cada uno de los vectores de su dominio obtenidos como salida en el proceso de entrenamiento. El vector que obtenga mayor similitud, será el designado como sentido desambiguado.

Sin embargo, al realizarse la competición SENSEVAL-2, aún no estaba publicado WordNet 1.7 y, por lo tanto, no había recursos etiquetados con dichos sentidos. La organización ofreció las tablas de actualización de los sentidos, cuyas correspondencias tenían cierta probabilidad de error. Así pues, debemos comprobar si la respuesta coincide con alguno de ellos en caso de que tenga más de una "traducción" en WordNet 1.7.

Por otro lado, las palabras del vector de entrada provocan la actualización de los pesos del vector ganador y nada impide que dicho vector ganador sea distinto en cada iteración, haciendo que la mayoría de los términos de los dominios figuren en muchos de los vectores del dominio, empequeñeciendo el valor de los pesos de estas palabras.

Para evitar este problema usamos la frecuencia normalizada como el peso de las palabras de los vectores de evaluación.

English-Lex-Sample		
Precisión	Recall	Coverage
0,598	0,597	99,741

Tabla 2: Resultados de la evaluación

En la Tabla 2, se observan los resultados de la evaluación. Dado que el corpus ofrecido por la organización fue diseñado específicamente para la competición, no es extraño que la cobertura sea prácticamente completa y, por ello, la precisión y recall son prácticamente iguales.

La red LVQ ha obtenido la octava mejor precisión y el sexto mejor recall de los 38 sistemas que compitieron en SENSEVAL-2.

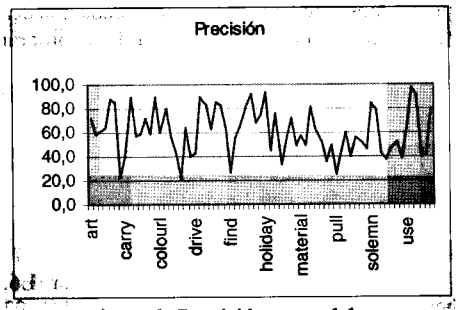

Figura 2: Precisión por palabra

En la Figura 2 se muestran las precisiones obtenidas por todas las palabras de la evaluación.

Un análisis del peor y mejor caso revela que la palabra "draw", verbo de 35 sentidos, es el peor caso con un 0,216 de precisión y un 0,216 de recall, mientras que la palabra "vital", adjetivo de 4 sentidos es el mejor caso con una precisión de 0,972 y un recall de 0,946.

7 *Conclusiones*

Hemos presentado un desambiguador basado en el MEV entrenado con SemCor y WordNet y

hemos simulado su participación en la competición SENSEVAL-2, concretamente en la tarea English-Lex-Sample. Se le ha incluido el corpus de entrenamiento que puso la organización a disposición de los competidores en su momento y se han calculado los pesos de las palabras con la red neuronal LVQ. El resultado de la simulación ha sido un octavo puesto en precisión con el sexto mejor recall.

La inclusión en los contextos de información morfosintáctica debe mejorar los resultados, puesto que se estarían separando, como palabras diferentes, las distintas ocurrencias de una misma palabra pero con distinto POS, enriqueciendo y diferenciando aún más los distintos vectores de un dominio.

Otra aportación fundamental para trabajos futuros consistiría en la introducción de técnicas estadísticas como la semántica latente, que permitirá destacar los términos importantes de cada dominio con lo que es de esperar que nuestro sistema mejore.

Por último, se deja para otra trabajo la aplicación a WSD de otras redes neuronales competitivas como la red de contrapropagación CPN (Counter Propagation Network) o los recientes modelos de la teoría de resonancia adaptativa ART (Adaptive Resonance Theory).

Los resultados obtenidos animan a participar en la próxima competición, SENSEVAL-3, en esta y otras modalidades de la competición para poder hacer una confrontación en igualdad de condiciones con el resto de sistemas.

Bibliografía

Allen, R. 1987. Several studies on natural language and backpropagation. *En Proc. of the First Annual Int. Conf. on NN.*

Atlas, L., R. Cole, J. Connor, M.El-Sharkawi, R.J. Maks, Y. Muthusamy, E. Barnard. 1989. Performance comparisons between backpropagation networks and classification trees on three real-word applications. En *Adv. In Neural Information Processing Systems*. Vol. 2, páginas 622-629.

Bishop, C.M. 1995. Neural networks for pattern recognition. Clarendon Press, Oxford.

Brown P., Della Pietra S., Della Pietra V. y Merce J., 1991.Word Sense Diambiguation Using Statistical Methods, Proceedings of ACL.

Bruce, R. y W. Janyce. 1994. Word sense disambiguation using decomposable models. En *Proceedings of 33rd Annual Meeting of the ACL.*

Cichocki, A., R. Unbehauen. 1993. Neural Networks for Optimization and Signal Processing. John Wiley.

Cottrell, G., S. Small. 1983. A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory*. Vol. 6. páginas 89-120

Daudé J., Padró L. and Rigau G. 2001. A Complete WN1.5 to WN1.6 Mapping, Proceedings of NAACL Workshop WordNet and Other Lexical Resources: Applications, Extensions and Customizations. Pittsburg, PA, United States,.

Escudero G., L. Márquez y G. Rigau. 2000. Boosting Applied to Word Sense Disambiguation. *European Conference on Machine Learning.*

Fellbaum, C. 1998. WordNet: An Electronic Lexical Database. The MIT Press

Fiesler, E., R. Beale. 1997. Handbook of Neural Computation. Oxford University Press.

Gonzalo J., Verdejo F., Chugur I, y Cigarrán J., 1998. Indexing WordNet synsets can improve text retrieval, en Proceedings of the ACL/COLING Workshop on Usage of WordNet for Natural Language Proceesing.

Ide, N y J. Veronis. 1998. Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics*. Vol. 24: 1.

Kilgarriff A. 1997. What is Word Sense Disambiguation Good for?, en Proceedings of Natural Language Processing Pacific Rim Symposium.

Kilgarriff A. 1998. SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. Proc. of LREC

Kilgarriff, A. y M. Palmer. 2000. Introduction to the special Sigue on SENSEVAL. Computers and the Humanities, 24 (1-2), 1-13.

Kohonen, T. 1995. Self-organization and associative memory. 2ª Edición, Springer.Verlag, Berlín.

Kohonen, T., J. Hynninen, J. Kangas, J. Laaksonen, K. Torkkola. 1996. Informe técnico, LVQ_PAK: The Learning Vector Quantization Program Package. Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finlandia.

Kohonen, T., S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, A. Saarela. 2000. Self oganizing of a massive document collection. *IEEE Transaction on neural networks*. Vol. 11, nº 3, páginas 574-585.

Martín, M.T., M. García, L.A. Ureña. 2002. Resolución de la ambigüedad mediante redes neuronales. *Procesamiento del Lenguaje Natural*, Revista nº 29.

Martín, M.T., M. García, L.A. Ureña. 2003. LVQ for text categorization using a multilingual linguistic resource. *Neurocomputing*. Pendiente de publicación.

Martínez, F., M.C. Díaz, M.T. Martín, V.M. Rivas, L.A. Ureña. 2002. Aplicación de redes neuronales y redes bayesianas en la detección de multipalabras para tareas IR. JOTRI 2002

McClelland, J., D. Rumelhart. 1986. Parallel Distributed Processing. Volúmenes I y II. MIT Press. Cambridge, MA.

Mihalcea, Rada y Moldovan. 2000. An iterative approach to word sense disambiguation, en Proceedings of Flair, pp. 219-233.Palomar M., M. Saiz-Noeda, R. Muñoz, A. Suarez , P. Martínez-Barco y A. Montoyo. 2000. PHORA: NLP System for Spanish. *Proc. CICLing 2001*. Mexico D.F.

Miller G., C. Leacock, T. Randee, R. Bunker. 1993. A Semantic Concordance. En *Proc. of the 3rd DARPA Workshop on Human Language Technology*.

Palomar M., M. Saiz-Noeda, R. Muñoz, A. Suarez , P. Martínez-Barco y A. Montoyo. 2000. PHORA: NLP System for Spanish. Proc. CICLing 2001. Mexico D.F.Robert, D., H: Moisl, H. Somers. 2000. *Handbook of Natural Language Processing. Marcel Dekker, Inc.*

Pedersen, P. y R Bruce. 1997. Distinguishing word senses in untagged text. En *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing.*

Ruiz, M.E., P. Srinivasan. 2002. Hierarchical text categorization using neural networks. Information Retrieval. 5, páginas 87-118.

Salton, G. y M.J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.

Schütze H., 1998. Automatic Word Sense Discrimination. Computational Linguistics, Vol 24, Nº 1, 1998.Towell, G., E.M. Voorhees. 1998. Disambiguating highly ambiguous words. *Computational Linguistic*. Vol. 24, nº1. pp. 125-145.

Shavlik, J.W., R.J. Mooney, G.G. Towell. 1991. Symbolic and neural learning algorithm: an experimental comparison. *Machine Learning*. Vol. 6, páginas 111-143.

Ureña L.A., Buenaga M. y Gómez J.M. 2001. Integrating Linguistic Resources in TC through WSD. *Computers and the Humanities*. vol. 35, nº 2.

Ureña López, L.A.; García Vega, M. y Martínez Santiago, F. 2001. Explotando las Relaciones Léxicas y Semánticas de WordNet en la Resolución de la Ambigüedad Léxica. VII Simposio Internacional de Comunicación Social.

Véronis, J., N.M. Ide. 1990. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. En *Proc. COLING-90*, páginas 389-394. Helsinki.

Widrow, B., S. D. Stearns. 1985. Adaptive signal processing. Signal processing series, Prentice-Hall.

Wilks Y. y M. Stevenson. 1998. Word sense disambiguation using optimised combinations of knowledge sources. *Proc. COLING-ACL*.

Xiaobin, L. y S. Spakowicz. 1995. WORDNET-based algorithm for word sense disambiguation. *Proc. of the Fourteenth International Joint conference on Artificial.*