Procesamiento del Lenguaje Natural, num. 29 (2002), pp.121-128 recibido 28-05-2002; aceptado 01-07-2002

Una formulación unificada para resolver distintos problemas de ambigüedad en PLN*

Antonio Molina, Ferran Pla, Encarna Segarra
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València (Spain)
{amolina,fpla,esegarra}@dsic.upv.es

Resumen: En este trabajo presentamos una formulación unificada de los Modelos de Markov Especializados. Estos modelos pueden ser aplicados a problemas de desambiguación en el procesamiento del lenguaje natural, siempre que estos problemas puedan abordarse como un proceso de etiquetado de los datos de entrada. La principal ventaja que presentan los Modelos de Markov Especializados, frente a los modelos de Markov básicos, radica en que permiten la incorporación de información adicional a los modelos. Para ilustrar el comportamiento de los modelos especializados se presenta un resumen de los resultados obtenidos en varias tareas de tratamiento de lenguaje natural.
Palabras clave: Técnicas basadas en corpus, modelos de Markov, etiquetado morfosintáctico, análisis sintáctico superficial, desambiguación semántica.

Abstract: In this work, we present an unified formulation of the specialized Markov models. These models can be applied to solve natural language disambiguation problems which can be considered as tagging problems. The main advantage of the specialized Markov models with respect to the basic Markov models is that they are able to incorporate additional information into de models. In order to show the behaviour of the specialized models we summarize the results obtained in different natural language processing tasks.
Keywords: Corpus-based techniques, hidden Markov models, Part-of-speech tagging, shallow parsing, word sense disambiguation.

1 Introducción

La disponibilidad actual de recursos lingüísticos, como corpora o diccionarios, ha hecho posible la utilización de aproximaciones inductivas o basadas en corpus en prácticamente todas las tareas de Procesamiento de Lenguaje Natural (PLN). El principal atractivo de estas técnicas es que generan modelos cuyos parámetros se estiman a partir de datos, y permiten un alto grado de modularidad y portabilidad.

Dentro de las aproximaciones inductivas podemos distinguir las basadas en máquinas de estados finitos y, en particular, los Modelos Ocultos de Markov. Estos modelos presentan, además de las características generales de las aproximaciones basadas en corpus, la ventaja de incorporar algoritmos muy eficientes tanto en la fase de entrenamiento de los modelos a partir de datos, como en el uso de estos modelos para el procesamiento de nuevos datos. Estos métodos han sido aplicados con éxito para resolver diferentes problemas en el tratamiento del lenguaje como son, por ejemplo, el reconocimiento automático del habla (Jelinek, 1997), el etiquetado morfosintáctico de textos –"POS tagging"– (Church, 1988) (Brants, 2000), (Pla y Molina, 2001), el análisis sintáctico superficial –"chunking"– (Molina y Pla, 2002) y la desambiguación del sentido de las palabras –"WSD"– (Loupy, El-Beze, y Marteau, 1998) (Molina et al., 2002).

Para poder llevar a cabo la desambiguación en PLN, utilizando Modelos de Markov, es necesario abordar cada una de las tareas como un problema de etiquetado. Este problema se puede formalizar de la siguiente manera. Sea $\mathcal{O}$ un conjunto de etiquetas y sea $\mathcal{I}$ el vocabulario de la aplicación. Dada una frase de entrada $I = i_1, \ldots, i_T$, el etiquetado

* Este trabajo ha sido parcialmente subvencionado por los proyectos CICYT TIC2000-0664-C02-01 y TIC2000-1599-C01-01

de la frase consiste en encontrar la secuencia de etiquetas de máxima probabilidad en el modelo. Para resolver esta maximización, se suelen introducir algunas simplificaciones (asunciones de Markov) que, aunque no siempre permiten proporcionar la solución exacta, permiten obtener resultados bastante precisos con costes computacionales aceptables. En particular, para Modelos de Markov de primer orden (bigramas) el problema de maximización se reduce a resolver la siguiente ecuación:

$$\arg\max_{o_1 \ldots o_T} \left(\prod_{j:1\ldots T} P(o_j|o_{j-1}) \cdot P(i_j|o_j) \right)$$

Los parámetros de esta ecuación se pueden representar como un Modelo de Markov en el que los estados tienen asociada una etiqueta del conjunto $\mathcal{O}$, las probabilidades de contexto, $P(o_j|o_{j-1})$, se corresponden con las probabilidades de transición entre estados del modelo, y $P(i_j|o_j)$ son las probabilidades de emisión de símbolos de entrada en los estados del modelo. El proceso de etiquetado se lleva a cabo de manera eficiente mediante programación dinámica utilizando el algoritmo de Viterbi.

Recientemente, se ha introducido una aproximación que usa Modelos de Markov a los que se incorpora la información disponible en los datos de entrenamiento: los Modelos de Markov Especializados (Pla, 2000). Esta incorporación de información se realiza mediante la redefinición de los datos del conjunto de entrenamiento. Por otra parte, aprovecha los mismos algoritmos que se usan en los modelos de Markov básicos para realizar la fase de entrenamiento de los modelos, y el análisis de los datos de entrada. Esta aproximación, que inicialmente fue aplicada a la tarea de etiquetado morfosintáctico para textos en inglés (Pla y Molina, 2001) y en castellano (Pla, Molina, y Prieto, 2001) obteniendo buenos resultados, ha sido posteriormente extendida con éxito para su aplicación a otras tareas de procesamiento de lenguaje natural: análisis superficial (Molina y Pla, 2002), detección de cláusulas (Molina y Pla, 2001) y desambiguación semántica (Molina et al., 2002).

En este artículo presentamos una formulación unificada de los Modelos de Markov Especializados y un resumen de las diferentes tareas de desambiguación en PLN que se han abordado bajo esta aproximación.

2 Modelos de Markov Especializados

Para construir Modelos de Markov Especializados que ofrezcan buenas prestaciones en la resolución de los problemas de ambigüedad anteriormente citados se deben tener en cuenta los siguiente aspectos:

- Seleccionar qué información disponible a la entrada es relevante para cada tarea. Por ejemplo, para el análisis sintáctico superficial podría ser suficiente considerar la etiqueta morfosintáctica pero, como hemos comprobado experimentalmente, si se consideran algunas palabras de la entrada se mejoran las prestaciones del modelo. A este proceso le llamaremos *selección*.
- Definir un conjunto de etiquetas de salida con la granularidad adecuada al problema y que evite la sobregeneralización que suelen presentar los modelos básicos de Markov. A veces, el conjunto de etiquetas para una determinada tarea, definido siguiendo criterios lingüísticos, es excesivamente reducido. Ello conlleva que el modelo aprendido sea demasiado general para producir buenos resultados, ya que el grado de ambigüedad es muy elevado. La redefinición de las etiquetas de salida, mediante la incorporación de la información de entrada disponible, ayuda a conseguir modelos que produzcan una menor sobregeneralización y, por lo tanto, sean más precisos. Por otra parte, ha de tenerse en cuenta que un número de etiquetas demasiado elevado puede dar lugar a problemas de estimación del modelo, sobretodo si no se dispone de datos de entrenamiento suficientes. Llamaremos *especialización* al proceso de redefinición del conjunto de etiquetas de salida.

A continuación se describe la técnica utiliza para construir los Modelos de Markov Especializados mediante la redefinición del conjunto de entrenamiento aplicando los procesos de *selección* y *especialización*.

Sean los conjuntos:

- $\mathcal{O} = \{o_1, o_2, \ldots, o_N\}$, el conjunto de etiquetas de salida.

- $\mathcal{R}_i = \{r_{i_1}, r_{i_2}, \ldots, r_{i_M}\}$, un rasgo de entrada (p.e. palabra, lema, etiqueta POS, etiqueta de 'chunk', sentido, etc.), donde los valores posibles de un rasgo (r_{i_j}) se consideran símbolos de un determinado alfabeto.
- $\mathcal{I} \subseteq \mathcal{R}_1 \times \mathcal{R}_2 \times \ldots \times \mathcal{R}_n$, el conjunto de tuplas de los n rasgos posibles de entrada.
- $\mathcal{T} \subset (\mathcal{I} \times \mathcal{O})^*$, el conjunto de entrenamiento.

Por ejemplo, si consideramos que $\mathcal{T} \subset (\mathcal{R}_1 \times \mathcal{R}_2 \times \mathcal{O})$, donde $\mathcal{R}_1$ es el conjunto de palabras (*Este, río, está, seco, ., ...*), $\mathcal{R}_2$ es el conjunto de etiquetas morfosintácticas (*DT, NC, V, ADJ, Fp, ...*) y $\mathcal{O}$ es el conjunto de etiquetas de "chunk" (*B-SN, I-SN, B-SV, B-SADJ, O, ...*), una muestra de $\mathcal{T}$ se representaría así:

Ejemplo 1:
⟨*Este, DT, B-SN*⟩ ⟨*río, NC, I-SN*⟩ ⟨*está, V, B-SV*⟩ ⟨*seco, ADJ, B-SADJ*⟩ ⟨*., Fp, O*⟩

La aplicación de una función f sobre el conjunto de entrenamiento original, $\mathcal{T}$, produce el nuevo conjunto de entrenamiento $\widetilde{\mathcal{T}} \subset (\widetilde{\mathcal{I}} \times \widetilde{\mathcal{O}})^*$:

$$f : \mathcal{T} \rightarrow \widetilde{\mathcal{T}}$$
$$f(< I_j, O_j >) =< f_S(I_j), f_E(I_j, O_j) >$$

donde $<I_j, O_j>$ es una tupla perteneciente a $(\mathcal{I} \times \mathcal{O})$ e $I_j =< r_{1_j}, r_{2_j}, ..., r_{n_j} >$ es un elemento de $\mathcal{I}$.

La función f consiste en la aplicación de la *función de selección* (f_S) sobre los rasgos de entrada y la *función de especialización* (f_E) sobre las etiquetas de salida.

2.1 Función de selección de los rasgos de entrada f_S

La función de selección, f_S, se define sobre el conjunto de tuplas de entrada, $\mathcal{I}$, y proporciona una nueva entrada, $\widetilde{\mathcal{I}}$:

$$f_S : \mathcal{I} \rightarrow \widetilde{\mathcal{I}}$$
$$f_S(< r_{1_j}, r_{2_j}, \ldots, r_{n_j} >) =$$
$$f_{S_1}(r_{1_j}) \cdot f_{S_2}(r_{2_j}) \cdot \ldots \cdot f_{S_n}(r_{n_j})$$

La nueva entrada, $\widetilde{\mathcal{I}}$, se forma mediante la operación de concatenación de cadenas (que se denota por $\cdot$) sobre aquellos rasgos que son significativos para una determinada tarea.

Para cada rasgo de entrada se considera un subconjunto $\mathcal{R}_{S_i} \subseteq \mathcal{R}_i$ que estará formado por los símbolos o valores relevantes para ese rasgo. Si todos los valores posibles para un rasgo determinado se consideran relevantes, entonces $\mathcal{R}_{S_i} = \mathcal{R}_i$.

Para rasgo, $\mathcal{R}_i$, se define una función, f_{S_i}, que determina si el valor del rasgo es relevante o no, es decir, si ese valor se concatenará o no a la entrada.

$$f_{S_i} : \mathcal{R}_i \rightarrow (\mathcal{R}_{S_i} \cup \lambda), \forall i : 1 \ldots n$$

$$f_{S_i}(r_{i_j}) = \begin{cases} r_{i_j} \text{ si } r_{i_j} \in \mathcal{R}_{S_i} \\ \lambda \text{ si } r_{i_j} \notin \mathcal{R}_{S_i} \end{cases}$$

De esta forma, si sobre el conjunto de entrenamiento mostrado en el Ejemplo 1 se aplica como criterio de *selección* que todas las etiquetas morfológicas son relevantes y del conjunto de palabras sólo consideramos la palabra *río*, es decir, $\mathcal{R}_{S_1} =$ { río } y $\mathcal{R}_{S_2} = \mathcal{R}_2$, el conjunto de entrenamiento ($\widetilde{\mathcal{T}}$) resultante sería:
⟨*DT, B-SN*⟩ ⟨*río·NC, I-SN*⟩ ⟨*V, B-SV*⟩ ⟨*ADJ, B-SADJ*⟩ ⟨*Fp, O*⟩

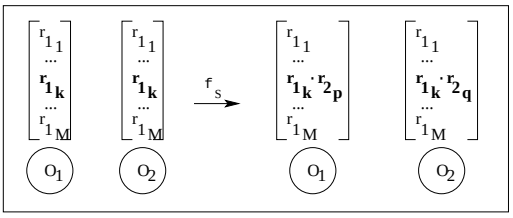

Figura 1: Efecto de la aplicación de la función de selección en los estados del modelo.

La aplicación de la función de selección permite incorporar al modelo cierto conocimiento determinado a priori que ayuda a resolver algunas ambigüedades. Por ejemplo, en la Figura 1 se puede observar que, en el modelo previo a la selección, el rasgo r_{1_k} puede ser emitido en los estados O_1 y O_2. Después de aplicar la función de selección, considerando como criterios de selección que $\mathcal{R}_{S_1} = \mathcal{R}_1$ y $\mathcal{R}_{S_2} = \{r_{2_p}, r_{2_q}\}$, dicho rasgo solamente puede ser emitido en el estado O_1 si se considera junto al rasgo r_{2_p} o en el estado O_2 si se considera junto al rasgo r_{2_q}.

2.2 Función de especialización de las etiquetas de salida f_E

La función de especialización, f_E, se define sobre las tuplas formadas por los rasgos de entrada, $\mathcal{I}$, y las etiquetas de salida $\mathcal{O}$, y proporciona un nuevo conjunto de etiquetas de salida, $\widetilde{\mathcal{O}}$, que es el resultado de redefinir las etiquetas de salida (o un subconjunto de ellas) añadiendo información disponible en la entrada.

$$f_E : \mathcal{I} \times \mathcal{O} \rightarrow \widetilde{\mathcal{O}}$$

$$f_E(< r_{1_j}, r_{2_j}, \ldots, r_{n_j}, o_j >) =$$

$$f_{E_1}(r_{1_j}) \cdot f_{E_2}(r_{2_j}) \cdot \ldots \cdot f_{E_n}(r_{n_j}) \cdot f_{E_{n+1}}(o_j)$$

De forma similar a como se define la función de selección, para cada rasgo de entrada se considera un subconjunto $\mathcal{R}_{E_i}$ que incluirá los símbolos relevantes para especializar la etiqueta de salida. Si todos los valores posibles para un rasgo determinado se consideran relevantes, entonces $\mathcal{R}_{E_i} = \mathcal{R}_i$.

Para cada rasgos, $\mathcal{R}_i$, se define una función, f_{E_i}, que determina si el valor se utiliza para redefinir la etiqueta de salida o no.

$$f_{E_i} : \mathcal{R}_i \rightarrow (\mathcal{R}_{E_i} \cup \lambda), \forall i : 1 \ldots n$$

$$f_{E_i}(r_{i_j}) = \begin{cases} r_{i_j} \text{ si } r_{i_j} \in \mathcal{R}_{E_i} \\ \lambda \text{ si } r_{i_j} \notin \mathcal{R}_{E_i} \end{cases}$$

Además, el conjunto de etiquetas de salida puede redefinirse añadiendo cierta información conocida sobre la entrada o los datos de entrenamiento. Esto se representa mediante la función $f_{E_{n+1}} : \mathcal{O} \rightarrow \mathcal{O}'$. Aunque, por lo general, se considera que $f_{E_{n+1}}(o_j) = o_j$.

Si sobre el Ejemplo 1, además de la selección anteriormente aplicada, se considera como criterio de *especialización* $\mathcal{R}_{E_1} = \{\}$ y $\mathcal{R}_{E_2} = \mathcal{R}_2$, el conjunto de entrenamiento resultante sería el siguiente:

⟨*DT, DT·B-SN*⟩ ⟨*río·NC, NC·I-SN*⟩ ⟨*V, V·B-SV*⟩ ⟨*ADJ, ADJ·B-SADJ*⟩ ⟨*Fp, Fp·O*⟩

En la Figura 2 se puede observar el efecto producido sobre el modelo contextual cuando se aplica la función de especialización. En este caso se ha considerado como criterio de especialización que $\mathcal{R}_{E_1} = \{r_{1_k}\}$. En el nuevo modelo aparece un nuevo estado por cada símbolo especializado y por cada uno de los estados que pueden emitir dicho símbolo. En este nuevo estado, $r_{1_k} \cdot O_1$, solamente se puede emitir el símbolo especializado, por lo que

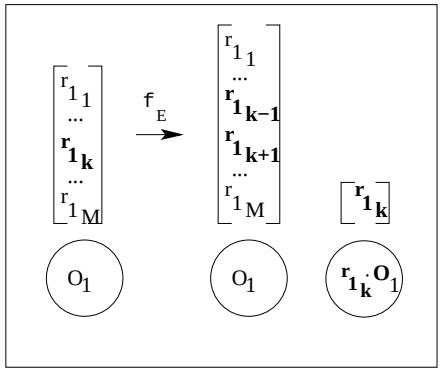

Figura 2: Efecto de la aplicación de la función de especialización sobre un estado del modelo.

la probabilidad de emisión será siempre uno. Esto permite modelizar un contexto particular para el símbolo escogido (r_{1_k}) en una determinada etiqueta de salida (O_1).

2.3 Ejemplos de Modelos de Markov Especializados

A continuación se muestra el efecto producido sobre los modelos de Markov aprendidos siguiendo el proceso de especialización descrito anteriormente.

Sean los siguientes conjuntos de rasgos de entrada $\mathcal{R}_1$ y $\mathcal{R}_2$ y sea $\mathcal{O}$ el conjunto de etiquetas de salida.

- $\mathcal{R}_1$={a,b,c,d}
- $\mathcal{R}_2$={x,y,z}
- $\mathcal{O}$={L,M}

En la Tabla 1 se muestran el conjunto de entrenamiento original $\mathcal{T} \subset (\mathcal{I} \times \mathcal{O})^*$ y los conjuntos de entrenamiento que se obtienen a partir de $\mathcal{T}$ aplicando los siguientes criterios de selección y especialización:

- $\widetilde{\mathcal{T}_1}$: $\mathcal{R}_{S_1} = \mathcal{R}_1, \mathcal{R}_{S_2} = \{\}, \mathcal{R}_{E_1} = \{\}, \mathcal{R}_{E_2} = \{\}$. Este conjunto de entrenamiento da lugar al modelo de Markov básico.
- $\widetilde{\mathcal{T}_2}$: $\mathcal{R}_{S_1} = \mathcal{R}_1, \mathcal{R}_{S_2} = \{\}, \mathcal{R}_{E_1} = \{a\}, \mathcal{R}_{E_2} = \{\}$. En el modelo correspondiente se han especializado las etiquetas de salida asociadas al símbolo de entrada a.
- $\widetilde{\mathcal{T}_3}$: $\mathcal{R}_{S_1} = \mathcal{R}_1, \mathcal{R}_{S_2} = \mathcal{R}_2, \mathcal{R}_{E_1} = \{\}, \mathcal{R}_{E_2} = \{\}$. En este caso, se seleccionan los dos rasgos disponibles en la entrada.

$\mathcal{T}$			$\widetilde{\mathcal{T}}_1$		$\widetilde{\mathcal{T}}_2$		$\widetilde{\mathcal{T}}_3$	
$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{O}$	$\tilde{\mathcal{I}}$	$\tilde{\mathcal{O}}$	$\tilde{\mathcal{I}}$	$\tilde{\mathcal{O}}$	$\tilde{\mathcal{I}}$	$\tilde{\mathcal{O}}$
b	x	L	b	L	b	L	b·x	L
a	x	M	a	M	a	a·M	a·x	M
c	z	L	c	L	c	L	c·z	L
b	y	M	b	M	b	M	b·y	M
a	x	L	a	L	a	a·L	a·x	L
b	y	M	b	M	b	M	b·y	M
c	z	L	c	L	c	L	c·z	L
d	x	M	d	M	d	M	d·x	M
a	x	L	a	L	a	a·L	a·x	L
b	y	M	b	M	b	M	b·y	M
d	z	M	d	M	d	M	d·z	M
a	x	L	a	L	a	a·L	a·x	L
d	z	M	d	M	d	M	d·z	M
d	x	M	d	M	d	M	d·x	M
c	z	L	c	L	c	L	c·z	L

Tabla 1: Distintos conjuntos de entrenamiento de ejemplo.

Los modelos que se muestran a continuación se corresponden con los Modelos de Markov de primer orden que se estimarían a partir del conjunto de entrenamiento correspondiente definido en la Tabla 1. En estos modelos se representan las probabilidades de emisión asociadas a cada estado y las probabilidades de transición entre estados, calculadas ambas a partir de las frecuencias de aparición en la muestra de entrenamiento correspondiente. Los estados etiquetados con <s> y </s> se corresponden con los estados inicial y final del modelo, respectivamente. Para facilitar la explicación de los ejemplos no se van a considerar las transiciones de suavizado que deberían aparecer en todos estos modelos. A continuación se analiza el comportamiento de cada uno de los modelos respecto a la cadena de entrada *b a c*, que se corresponde con una muestra vista en el corpus de entrenamiento.

A partir del conjunto de entrenamiento $\widetilde{\mathcal{T}}_1$ se aprende el modelo básico de la Figura 3. Para la cadena *b a c* existen dos caminos de análisis en el modelo: *<s> L M L </s>* y *<s> M M L </s>*. La elección del mejor camino vendrá determinada por las distribuciones de probabilidad del modelo. Además, se puede observar también que el modelo reconoce todas las cadenas que se pueden formar con los símbolos de entrada, excepto aquellas que tengan más de una *c* consecutiva.

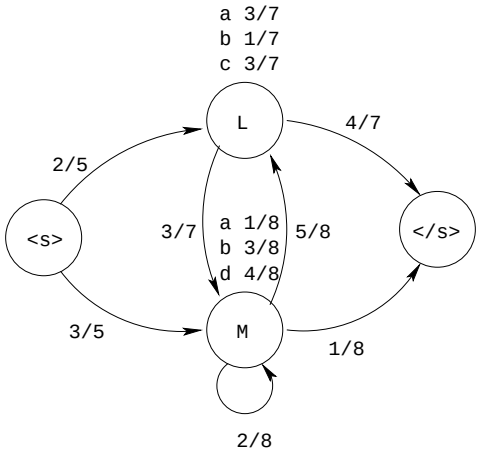

Figura 3: Modelo básico obtenido a partir del conjunto de entrenamiento $\widetilde{\mathcal{T}}_1$.

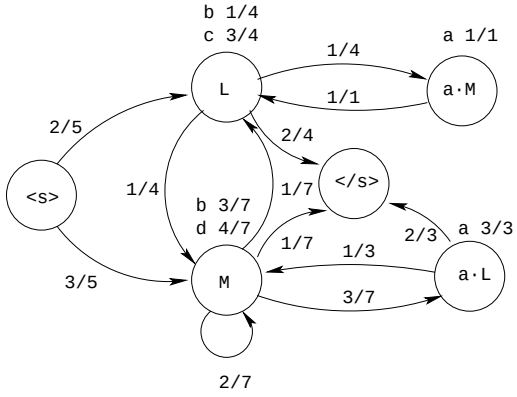

Figura 4: Modelo especializado obtenido a partir del conjunto de entrenamiento $\widetilde{\mathcal{T}}_2$.

En el modelo especializado obtenido a partir del conjunto de entrenamiento $\widetilde{\mathcal{T}}_2$ (Figura 4), en el cual se ha elegido el símbolo *a* para especializar el modelo, se puede observar que:

1. Aparecen dos nuevos estados $a \cdot L$ y $a \cdot M$ que emiten el símbolo *a* de manera que la modelización contextual se hace más rica. Por ejemplo, el símbolo *a* sólo puede etiquetarse con *M* en un determinado contexto.
2. Este modelo especializado no admite cadenas con más de una *a* consecutiva, que sí que eran reconocidas por el modelo básico (y al igual que en el modelo básico, tampoco admite cadenas con más de una *c* consecutiva).
3. En este caso sólo existe un camino posible para la cadena *b a c*. Este camino se

corresponde con la secuencia de etiquetas de salida correcta según los datos de entrenamiento.

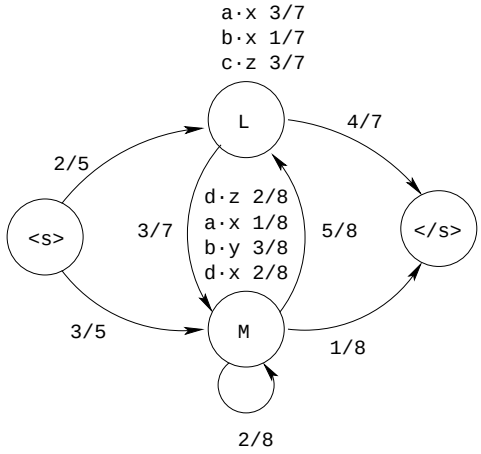

Figura 5: Modelo con selección obtenido a partir del conjunto de entrenamiento $\widetilde{\mathcal{T}}_3$.

La selección de rasgos de entrada también permite resolver ciertas ambigüedades. En el conjunto de entrenamiento $\widetilde{\mathcal{T}}_3$ se han seleccionado todos los símbolos de los dos rasgos de entrada, $\mathcal{R}_1$ y $\mathcal{R}_2$. El modelo aprendido, Figura 5, presenta la misma topología que el modelo básico distinguiéndose de éste en los símbolos emitidos en los estados. El símbolo b se emite en el estado L si va acompañado del símbolo x, y en el estado M si va acompañado de y. Para la cadena $b\ a\ c$, que en este caso sería $b{\cdot}x\ a{\cdot}x\ c{\cdot}z$, sólo existe un camino posible en el modelo que se corresponde con la secuencia de etiquetas de salida correcta según los datos de entrenamiento.

Estos ejemplos ilustran cómo se pueden obtener modelos más adaptados a las muestras de entrenamiento y menos afectados por la sobregeneralización que suele presentar el modelo básico.

3 Descripción del proceso de aprendizaje

El proceso de aprendizaje de un Modelo de Markov Especializado es similar al aprendizaje de un Modelo de Markov básico. La única diferencia estriba en que para entrenar los Modelos Especializados se parte de una adecuada redefinición de los datos de entrenamiento. Un Modelo de Markov Especializado se obtiene según los siguientes pasos (ver Figura 6):

Tarea	Criterio de Selección
POS Tagging	$\mathcal{R}_{S_{PAL}} = \mathcal{R}_{PAL}$
Chunking	$\mathcal{R}_{S_{PAL}}$ = {palabras cuyo ratio de error es mayor que 2} ∪ {palabras cuyo chunk asociado es SBAR, PP o VP} $\mathcal{R}_{S_{POS}} = \mathcal{R}_{POS}$
Clausing	$\mathcal{R}_{S_{PAL}}$ = {palabras de frecuencia alta} $\mathcal{R}_{S_{POS}} = \mathcal{R}_{POS}$ $\mathcal{R}_{S_{CH}} = \mathcal{R}_{CH}$
WSD	$\mathcal{R}_{S_{LEMA}} = \mathcal{R}_{LEMA}$ $\mathcal{R}_{S_{POS}} = \mathcal{R}_{POS}$

Tabla 2: Mejores criterios de selección para cada tarea.

1. Se define la información de entrada relevante para la tarea (*criterio de selección*).
2. Se redefinen las etiquetas de salida utilizando aquella información de la entrada que sea relevante (*criterio de especialización*).
3. Se aplican los criterios escogidos al conjunto de entrenamiento original para producir uno nuevo.
4. Se aprende un modelo especializado a partir del nuevo conjunto de entrenamiento.
5. Se desambigua el conjunto de desarrollo utilizando el modelo aprendido.
6. Se evalúa la salida del sistema para contrastar el comportamiento de los criterios aplicados sobre el conjunto de desarrollo.

Estos pasos se realizan combinando de diversas formas la información de entrada para determinar los mejores criterios de selección y especialización. En la Tabla 2 y la Tabla 3 se resumen los criterios con los cuales el sistema ha ofrecido mejores prestaciones para cada una de las tareas de desambiguación abordadas.

4 Resultados experimentales y conclusiones

En esta sección se describen los experimentos realizados sobre las tareas de etiquetado morfosintáctico ("POS tagging"), análisis sintáctico superficial ("chunking"), detección

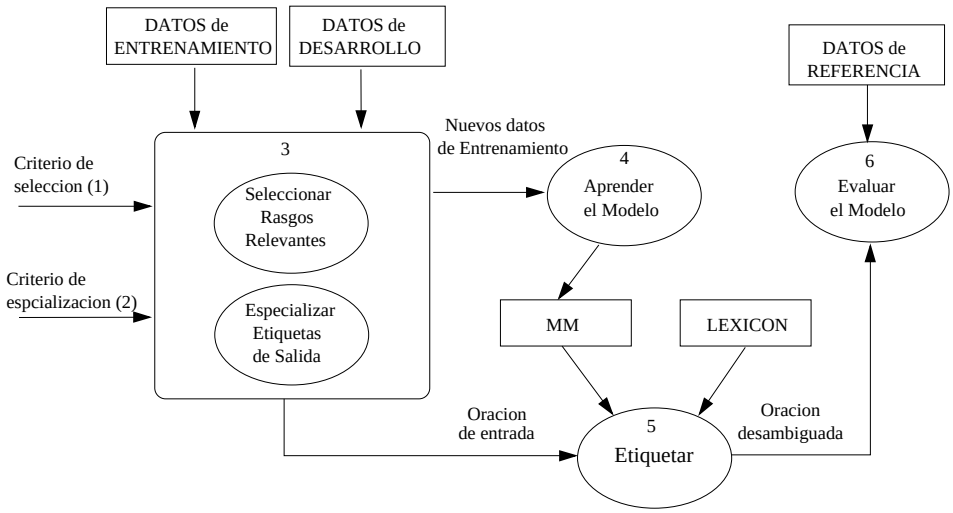

Figura 6: Esquema del proceso de aprendizaje.

Tarea	Criterio de Especialización
POS Tagging	$\mathcal{R}_{E_{PAL}}$ = {palabras cuya frecuencia de aparición es mayor que 2000 } $\mathcal{O} = \mathcal{R}_{POS}$
Chunking	$\mathcal{R}_{E_{PAL}}$ = {palabras cuyo ratio de error es mayor que 2} ∪ {palabras más frecuentes cuyo chunk asociado es SBAR, PP o VP} $\mathcal{R}_{E_{POS}} = \mathcal{R}_{POS}$ $\mathcal{O} = \mathcal{R}_{CH}$
Clausing	$\mathcal{R}_{E_{PAL}}$ = {palabras de frecuencia alta} $\mathcal{R}_{E_{POS}} = \mathcal{R}_{POS}$ $\mathcal{R}_{E_{CH}} = \{\}$ $\mathcal{O}$ = etiquetas de cláusula enumeradas según los niveles de anidamiento
WSD	$\mathcal{R}_{E_{LEMA}}$ = {lemas que aparecen en WordNet cuya frecuencia de aparición es mayor que 20} ∪ {lemas que no aparecen en WordNet} $\mathcal{O}$ = {sentidos de WordNet}

Tabla 3: Mejores criterios de especialización para cada tarea.

de cláusulas ("clausing") y desambiguación del sentido de las palabras ("WSD").

Los datos de entrenamiento y de prueba utilizados han sido los disponibles para el inglés (ya que no se disponen de corpora etiquetados para el castellano): para las tareas de etiquetado morfosintáctico y análisis sintáctico ("chunking" y detección de cláusulas), se ha utilizado la parte etiquetada del corpus *Wall Street Journal* (WSJ) en el proyecto Penn TreeBank (versión 2.0). Para la tarea de desambiguación semántica se ha

Tarea	Precisión	Cobertura
POS Tagging	96.8%	96.8%
Chunking	92.0%	92.4%
Clausing	70.9%	65.6%
WSD	60.2%	60.2%

Tabla 4: Resultados de precisión y cobertura para cada tarea.

utilizado el corpus *SemCor* [1] como conjunto de datos de aprendizaje y el corpus proporcionado en la última edición de *Senseval-2* [2] como conjunto de datos de prueba. Los resultados se evalúan en términos de precisión[3] y cobertura[4].

El comportamiento de nuestro sistema para las distintas tareas mostradas en la Tabla 4 es satisfactorio, obteniéndose unos resultados que son comparables a los de las aproximaciones más relevantes en estos campos.

En una tarea de etiquetado morfosintáctico, sobre el corpus Wall Street Journal, se alcanza una precisión del 96.8% que es comparable al obtenido por otras aproximaciones bajo las mismas condiciones experimentales: 96.9% con un modelo de máxima entropía, 96.5% con el etiquetador

[1] El corpus *SemCor* se encuentra disponible en http://www.cogsci.princeton.edu/~wn/

[2] Los datos para aprendizaje y evaluación utilizados en *Senseval-2* están disponibles en http://www.sle.sharp.co.uk/senseval2/

[3] Precision(P) = # Unidades correctamente etiquetadas / # Unidades etiquetadas

[4] Cobertura(C) = # Unidades correctamente etiquetadas / # Unidades a etiquetar

de Brill y 96.5% con la aproximación basada en "memory-based learning" (Pla y Molina, 2001).

En el problema de análisis sintáctico superficial se ha hecho una exhaustiva comparación con diferentes aproximaciones usando distintos paradigmas (Tjong Kim Sang y Buchholz, 2000) (Molina y Pla, 2002). De los doce sistemas comparados, las mejores prestaciones, medidas en términos del factor F_β[5], son alcanzadas por sistemas combinados (93.9) y las más bajas por los sistemas basados en reglas (87.2). De estos sistemas, cinco superan el valor F_β=92.2 que es el resultado de nuestra aproximación.

En la tarea de detección de cláusulas, se ha realizado una comparación con cinco sistemas (Tjong Kim Sang y Déjean, 2001) (Molina y Pla, 2001). Nuestro sistema, que obtiene un F_β=68.1, se encuentra en segundo lugar, por debajo de la aproximación basada en el método de aprendizaje "Ada-Boost".

El valor de precisión obtenido sobre la tarea "English all-words" de Senseval-2 situaría a nuestro sistema (60.2%) en cuarto lugar. Los valores de precisión de los 3 sistemas mejores son respectivamente, 69.0% (SMUaw), 63.6% (CNTS-Antwerp) y 61.8% (SinequaLIA).

Según estos resultados, podemos destacar el buen comportamiento de los Modelos de Markov Especializados en las tareas que permiten una formulación en términos de etiquetado. Además de presentar las ventajas inherentes a las aproximaciones basadas en corpus, y en particular, de las basadas en modelos de Markov, permiten obtener unos modelos más adaptados a los datos de entrenamiento, y por tanto a la tarea, a través de la definición de unos buenos criterios de *selección* y *especialización*.

Bibliografía

Brants, Thorsten. 2000. TnT – a statistical part-of-speech tagger. En *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA.

Church, K. W. 1988. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. En *Proceedings of the 1st Conference on Applied Natural Language Processing, ANLP*, páginas 136–143. ACL.

Jelinek, F. 1997. *Statistical Methods for Speech Recognition*. The MIT Press.

Loupy, C., M. El-Beze, y P. F. Marteau. 1998. Word Sense Disambiguation using HMM Tagger. En *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, páginas 1255–1258, Granada, Spain, May.

Molina, Antonio y Ferran Pla. 2001. Clause detection using HMM. En *Proceedings of the 5th Conference on Computational Natural Language Learning (CoNLL-2001)*, Toulouse, France, July.

Molina, Antonio y Ferran Pla. 2002. Shallow Parsing using Specialized HMMs. *Journal of Machine Learning Research*, 2:595–613.

Molina, Antonio, Ferran Pla, Encarna Segarra, y Lidia Moreno. 2002. Word Sense Disambiguation using Statistical Models and WordNet. En *Proceedings of 3rd International Conference on Language Resources and Evaluation, LREC2002*, Las Palmas de Gran Canaria, Spain.

Pla, Ferran. 2000. *Etiquetado Léxico y Análisis Sintáctico Superficial basado en Modelos Estadísticos*. Phd. Thesis, Dep. de Sistemes Informàtics i Computació. Universitat Politècnica de Valéncia.

Pla, Ferran y Antonio Molina. 2001. Part-of-Speech Tagging with Lexicalized HMM. En *proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP2001)*, Tzigov Chark, Bulgaria, September.

Pla, Ferran, Antonio Molina, y Natividad Prieto. 2001. Evaluación de un etiquetador morfosintáctico basado en bigramas especializados para el castellano. *Revista para el Procesamiento del Lenguaje Natural*.

Tjong Kim Sang, Erik F. y Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. En *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September.

Tjong Kim Sang, Erik F. y Hervé Déjean. 2001. Introduction to the CoNLL-2001 shared task: Clause identification. En *Proceedings of the 5th Conference on Computational Natural Language Learning (CoNLL-2001)*, Toulouse, France, July.

[5] $F_{\beta=1} = \frac{2 \times P \times C}{P + C}$