Análisis e implementación de diferentes técnicas de etiquetado de texto en español

Pedroza Méndez Blanca Estela[1], Cárcamo Martínez Lourdes[2]

Instituto Tecnológico de Apizaco
Maestría en Ciencias en
Ciencias de la Computación
División de estudios de posgrado
Av. Instituto Tecnológico s/n.
Apizaco Tlaxcala, México.
[1]thelismedina@hotmail.com, [2]lysa_c@hotmail.com
Paper received on 12/08/08, accepted on 06/09/08.

Resumen. El presente trabajo muestra y analiza los resultados de una arquitectura para el etiquetado de palabras en español, basada en los algoritmos de Viterbi y el ,de frecuencias relativas que utilizan modelos ocultos de Markov, además analizamos algunos resultados de la implementación del algoritmo chart parser. Para la ejecución de las técnicas de etiquetado utilizamos dos corpus textuales que son el CONLL y CLIC-TALP. Las pruebas que se efectuaron se realizaron tomando un noventa por ciento del texto de los corpus para cuerpo de entrenamiento de los etiquetadores y un diez por ciento para probarlos. Una vez finalizado el entrenamiento de los diferentes corpus y la ejecución de cada una de las técnicas así como de la arquitectura propuesta; se procede a la recopilación de las etiquetas que arrojaron las diferentes técnicas y las etiquetas predichas que conforman el 10% del corpus para prueba, estas etiquetas las sometemos a un análisis mediante un programa de evaluación utilizando métricas de efectividad, teniendo los porcentajes de "Recall", "Precision", "Error", "Accuracy" y "FB_1".

1 Introducción

La lingüística computacional es un área de las ciencias computacionales y su meta es desarrollar una teoría del lenguaje computacional, usando las nociones de algoritmos y estructura de datos [7]. Uno de los objetivos de la Lingüística Computacional es incorporar el conocimiento lingüístico a través de la simulación por la computadora. Dentro de la lingüística computacional podemos encontrar un área conocida como Procesamiento del Lenguaje Natural o PLN y uno de los objetivos que persigue es el perfecto análisis y entendimiento de los lenguajes humanos [5]. Los esfuerzos de investigación en PLN han sido dirigidos hacia tareas intermedias que dan sentido a alguna de las múltiples características estructurales inherentes a los lenguajes, sin requerir un entendimiento completo. Una de esas tareas es la asignación de categorías gramaticales a cada una de las palabras del texto. Este proceso se denomina también etiquetación de las partes del discurso o POS [5].

El proceso de etiquetación debe eliminar ambigüedades y encontrar cual es el papel más probable que juega cada palabra dentro de una frase. Dicho proceso debe

© M. G. Medina Barrera, J. F. Ramirez Cruz, J.H. Sossa Azuela, (Eds.).
Advances in Intelligent and Information Technologies
Research in Computing Science 38, 2008, pp 237-251.

ser capaz también de asignar una etiqueta a cada una de las palabras que aparecen en un texto y que no están presentes en nuestro diccionario, y garantizar de alguna manera que ésa es la etiqueta correcta.

El problema de la etiquetación se aborda tradicionalmente a partir de recursos lingüísticos bajo la forma de diccionarios y textos escritos, previamente etiquetados o no. Esta línea de desarrollo es la que se denomina lingüística basada en corpus. Dichos textos se utilizan para ajustar los parámetros de funcionamiento de los etiquetadores. Este proceso de ajuste se denomina entrenamiento. Las técnicas tradicionales engloban métodos estocásticos, tales como los modelos de Markov ocultos aunados al algoritmo de Viterbi, el cual es un método para asignar categorías gramaticales [1].

Por tanto, uno de los objetivos del presente trabajo consiste en implementar una de las herramientas de etiquetación, en específico una herramienta estocástica que permita integrar información específica para el español, y posteriormente realizar una evaluación exhaustiva de este y otros modelos.

Aun cuando el porcentaje de etiquetación en la mayoría de los etiquetadores existentes en la actualidad es de 97 – 98%, el pequeño porcentaje de palabras etiquetadas erróneamente (2-3%) es una característica que está siempre presente en los sistemas de etiquetación puramente estocásticos. Por esta razón, apoyamos la idea del uso de estos sistemas en combinación con información sintáctica, esto es, con técnicas de análisis sintáctico, y éste es otro de los objetivos del presente trabajo. Para esto utilizaremos un analizador sintáctico llamado Chart Parser.

La estrategia consiste en combinar fragmentos o subsecuencias de etiquetas para generar varias etiquetaciones completas posibles de la frase en cuestión, y posteriormente aplicar un filtro estadístico para elegir la secuencia de estados más probable. Para cumplir este objetivo, es imprescindible la disponibilidad de una gramática, un corpus de entrenamiento y un diccionario previamente etiquetado.

2 Desarrollo del modelo

En la figura 1 se muestra la arquitectura que proponemos para combinar los diferentes etiquetadores, en esta se muestra que tenemos como primer paso la obtención de una oración del texto a etiquetar, la sometemos al analizador sintáctico "Chart Parser", en el que a través de un diccionario de datos y una gramática, se determina si una oración es válida o no de acuerdo a su estructura gramatical. Si la oración es válida el siguiente paso es determinar si existe o no ambigüedad. Para esto se verifica si en cada palabra de la oración existe mas de una posible etiqueta que se le podría asignar, para el caso en el que exista ambigüedad la oración se someterá a un nuevo etiquetado con los Modelos de Markov Ocultos a través del uso de Frecuencias Relativas para determinar la secuencia de etiquetas óptima, el resultado que arroje se interpretará como una de las siguientes opciones: confirmación o mejora del etiquetado que dio con el analizador Chart Parser, o asignar un etiquetado erróneo; con el resultado que haya arrojado, se da por terminado el etiquetado de esa oración y se continua con la siguiente. Para el caso en el que no exista ambigüedad se continúa con la siguiente oración del texto. Regresando al paso en el que se determina si una oración es válida o no, continuamos con la opción de la oración no válida, esta se

someterá a un nuevo etiquetado con el Algoritmo de Viterbi, el resultado que arroje se tomará como una de las siguientes opciones: confirmación o mejora del etiquetado que dio con el analizador Chart Parser, o asignación de un etiquetado erróneo; con el resultado que haya arrojado, se da por terminado el etiquetado de esa oración y se continúa con la siguiente.

La entrada para cualquiera de las técnicas de etiquetado es algún texto no restringido, el cual es separado en unidades o tokens. Por cada token detectado, se obtiene sus posibles etiquetas a través del uso de un diccionario.

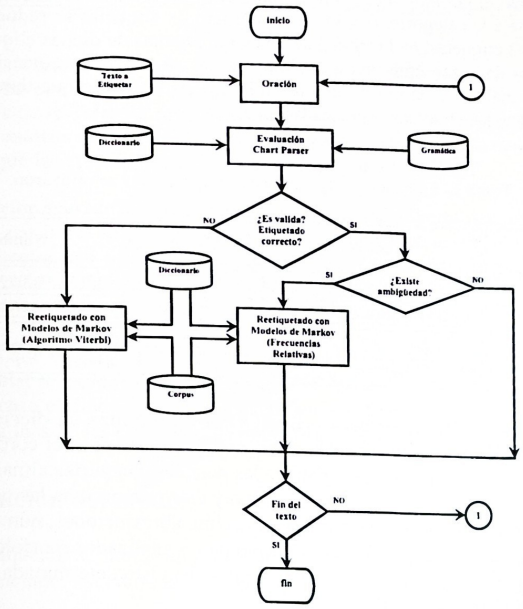

Fig.1. Esquema de la arquitectura propuesta

Para este trabajo se definieron varias etiquetas conforme a la morfología de la gramática del lenguaje español, las cuales se utilizaron para el etiquetado de textos durante la programación, en la tabla 1 se muestran algunas.

2.1 Componentes de los etiquetadores

2.1.1. Corpus CONLL. El CoNLL-2002 entidades nombradas consta de archivos con datos que abarcan dos idiomas: español y holandés [7]. Las etiquetas utilizadas en este corpus son las definidas en el proyecto PAROLE. Este conjunto consta de unas 230 etiquetas, estructuradas en categorías y subcategorías, y que contemplan diferentes aspectos como género, número, tiempo verbal, persona, etc.

Debido al reducido número de muestras de aprendizaje supervisadas disponibles se ha reducido el conjunto completo de etiquetas considerando solamente información referente a la categoría, con lo que el número de etiquetas se reduce considerablemente a 62 etiquetas, en la tabla 2 se muestran algunas de dichas etiquetas.

Para usos de la presente investigación se optó por utilizar abreviaciones encontradas en el análisis de la morfología del idioma español, pero que tienen el mismo significado que las etiquetas encontradas en el Corpus CONLL [3].

Tabla 1. Ejemplos de algunas de las etiquetas que se utilizaron.

Etiqueta	Nombre	Etiqueta	Nombre
Adj_cal	adjetivo calificativo	pron_num	pronombre numeral
Adj_ord	adjetivo ordinal	pron_exc	pronombre exclamativo
Adv	Adverbio	pron	pronombre
Adv_neg	adverbio negative	num	numeral

2.1.1.1 Diccionario de datos del Corpus CONLL. Se utiliza un diccionario construido a partir del corpus que se esté considerando, en este caso el corpus CONLL, el diccionario colecciona para cada palabra las posibles categorías almacenadas. Para cada unidad de entrada se consulta el mismo y éste proporciona las posibles categorías. Es capaz de identificar signos de puntuación, abreviaciones, números, etc. En éstas condiciones, el diccionario hace el papel de un analizador morfológico que es capaz de proporcionar todas las etiquetas en que se ha visto determinada palabra, pero restringido al corpus utilizado.

En este caso el diccionario esta conformado por 28518 palabras, en la tabla 2 se muestran algunas de las etiquetas que lo conforman y su respectiva cantidad.

2.1.2 Corpus CLiC-TALP. CLiC-TALP es un subcorpus creado a partir de LexEsp, constituido para ser utilizado como base para el aprendizaje automático para la desambiguación morfosintáctica [2].

El corpus CLiC- TALP proviene de dos fuentes diferentes. Por una parte recoge una muestra representativa de un corpus de prensa de 7 millones de palabras proporcionado por el periódico "La Vanguardia". Por otra, recoge una muestra del corpus LexEsp (Léxico informatizado del español). Es representativo del español estándar escrito porque presenta varios estilos narrativos, procedentes de distintas fuentes (literatura, prensa, etc.) e incluye también muestras tanto del español peninsular como del de América [2].

Tabla 2. Ejemplos de algunas etiquetas que integran el diccionario de datos del corpus CONLL.

Etiquetas	Cantidad	Etiquetas	Cantidad	Etiquetas	Cantidad
adj_cal	6571	vbo_part	873	sig_tp	2
adj_ord	58	vbo_aux	38	sig_pvc	1
adv	363	vbo_cop	30	sig_dp	1
adv_neg	2	interj	9	sig_c	5

El corpus CLiC-TALP consta actualmente de 100.000 palabras anotadas manualmente a nivel morfosintáctico [2].

Para usos de la presente investigación se optó por utilizar abreviaciones encontradas en el análisis de la morfología del idioma español, pero que tienen el mismo significado que las etiquetas encontradas en el Corpus CLiC-TALP.

2.1.2.1. Diccionario de datos del corpus CLiC-TALP. Se utiliza un diccionario construido a partir del corpus que se está considerando, en este caso el corpus CLiC-TALP, el diccionario colecciona para cada palabra las posibles categorías almacenadas. Para cada unidad de entrada se consulta el mismo y éste proporciona las posibles categorías. Es capaz de identificar signos de puntuación, abreviaciones, números, etc. En éstas condiciones, el diccionario hace el papel de un analizador morfológico que es capaz de proporcionar todas las etiquetas en que se ha visto determinada palabra, pero restringido al corpus utilizado.

En este caso el diccionario esta conformado por 19401 etiquetas, en la tabla 3 se muestran algunas de las etiquetas que lo conforman y su respectiva cantidad.

2.2. Definición del modelo

A continuación se muestran las diferentes técnicas que se utilizarán para el proceso de etiquetado, así como los procesos que se siguen en cada una de ellas.

Tabla 3. Ejemplos de etiquetas que integran el diccionario de datos del corpus CLiC-TALP.

Etiquetas	Cantidad	Etiquetas	Cantidad	Etiquetas	Cantidad
sust	6672	sig_pvc	1	sig_ac	1
sig_ic	1	conj_coo	26	sig_dp	1
prep	155	pron_pos	11	vbo_imp	95
pron_dem	32	vbo_part	409	interj	52
vbo_ind	2802	nom	1704	det_exc	2

2.2.1. Analizador Sintáctico. Uno de los procesos que se muestran en la figura 1, es el de un analizador sintáctico, en este caso el que utilizamos para esta investigación es el "Chart Parser", en él se reconocen cuáles oraciones son gramaticalmente acep-

tables y se les asigna una estructura sintáctica (etiquetas), este proceso se verifica a través de un conjunto de reglas de producción (gramática) y a través de una técnica de análisis sintáctico (parsing) usando una estrategia de búsqueda llamada bottom-up, su funcionamiento se inicia a partir de las palabras de una oración, posteriormente a través de las reglas de producción se hace un recorrido en reversa hasta llegar al símbolo inicial de la gramática.

2.2.2. Gramática. El tipo de gramática que se utiliza es el de una gramática libre del contexto, ésta es una descripción formal de la sintaxis de un lenguaje. En este caso el lenguaje al que nos referimos es el español, se realizó un análisis detallado de su morfología y sintaxis para generar las reglas de producción que más adelante se muestran.

Una gramática se define formalmente como [4] :

$$G = \{T, NT, S, P\}$$

Donde:

T = ***Conjunto de símbolos terminales***

Los símbolos terminales que utilizamos en el analizador sintáctico se muestran en la tabla 1, estos símbolos, se definieron de acuerdo a la morfología del idioma español y representan a cada una de las categorías gramaticales.

NT = ***Conjunto de símbolos no terminales.***

Los símbolos no terminales lo conforman aquellos que pueden ser descompuestos y formar a su vez nuevas reglas de producción. Algunos de los que utilizamos para este proyecto se muestran en la tabla 4 los cuales ayudan a conformar la gramática que utilizaremos en el analizador sintáctico.

S = ***Símbolo inicial que pertenece a NT.***

Las gramáticas tienen un símbolo especial llamado símbolo inicial, en este caso, el símbolo inicial que se ha definido para nuestra gramática es la letra O_r.

P = ***Conjunto de Reglas de producción.***

Las reglas de producción describen las diferentes estructuras que son permitidas en el idioma español desde la estructura mas pequeña (Det → det art) hasta la mas amplia (or → SN sig_com conj_coo sig_com SPred), éstas fueron definidas después de realizar un análisis profundo de la morfología del idioma español. En total se generaron 137 reglas de producción.

2.2.3.Elementos del Modelo de Markov. Un Modelo Oculto de Markov (HMM) es un modelo doblemente estocástico, ya que uno de los procesos no se puede observar directamente (está oculto), sino que se puede observar sólo a través de otro conjunto de procesos estocásticos, los cuales producen la secuencia de observaciones [5].

Tabla 4. Ejemplos de algunos símbolos No Terminales.

No Terminales	Significado	No Terminales	Significado
Or	Oración	SAdj	sintagma adjectival
I	Interjección	N	nombre
SN	sintagma nominal	Vbo	verbo
Spred	sintagma predicativo	vbo_aux	verbo auxiliary

Uno de los procesos de etiquetado con Modelos de Markov Ocultos es mediante el algoritmo de Viterbi. El proceso consiste en encontrar la secuencia de estados de mayor probabilidad. Una vez obtenida dicha secuencia, como cada estado tiene asociada una única categoría léxica se dispone de la mejor secuencia de categorías para la secuencia de palabras de entrada. Cabe recordar que cada estado es una etiqueta dentro del conjunto de etiquetas que se manejan en el corpus.

Un HMM se caracteriza por la 5-tupla (Q; V; π; A; B), donde:

- ✓ Q es el conjunto de estados del modelo, representándose cada estado por $q_1, q_2, ...q_N$ donde N es el total de estados. Para el desarrollo de este trabajo se contemplan los estados que se incluyen en el corpus de entrenamiento y que están definidos en la tabla 3, dado que son los que aparecen al menos una vez dentro del corpus.
- ✓ $\pi = \{\pi_i\}$ es la distribución de probabilidad del estado inicial. Formalmente se define como:

$$\pi_i = P(q_1 = i), \quad \pi_i \geq 0, \quad 1 \leq i \leq N, \quad \sum_{i}^{N} \pi_i = 1 \qquad (1)$$

Para obtener el vector de probabilidad del estado inicial se realizó lo siguiente: se obtuvo la suma total de cada estado inicial por oración y se dividió entre el número total de estados iniciales del corpus. Algunas de las probabilidades que se obtuvieron para el vector π se muestran en la tabla 5, ésta información generada se obtuvo del corpus CONLL y se utilizó en el algoritmo de Viterbi.

- ✓ A = {aij} es la distribución de probabilidad de las transiciones entre estados.

a_{ij} = probabilidad de que ocurra el estado q_i dado que ocurrio el estado qi

$$i = 1, 2, ..., N$$

Formalmente se define como:

$$a_{ij} = P(q_t = j / q_{t-1} = i) = p(j / i) \quad 1 \leq i, j \leq N \qquad (2)$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad \forall i. \qquad (3)$$

Tabla 5. Algunas probabilidades del estado inicial.

Estado	Probabilidad inicial	Estado	Probabilidad inicial

Nom	0.043921	pron	0.0092248
sig_pa	0.011703	vbo_sub	0.0035798
sig_pc	0.0015145	conj_coo	0.00013768

Para formar la matriz de transición entre estados, es necesario primero obtener los unigramas y bigramas de estados del corpus de entrenamiento. El proceso que se sigue para la obtención de los n-gramas del corpus de entrenamiento es el siguiente:

- Se lee desde archivo la información del corpus.
- Posteriormente se obtienen los unigramas y bigramas tanto de etiquetas como de palabras-etiquetas del cuerpo de entrenamiento.
- A partir de esto se obtienen las frecuencias y probabilidades tanto de unigramas como de bigramas para las dos categorías mencionadas anteriormente.
- Para obtener los unigramas se obtiene cada estado (q_i) que aparece en el corpus y para la frecuencia de unigramas se realiza la suma de cada estado (q_i) que aparece en el corpus.

El tamaño de la matriz de transición es de N por N estados. La condición para formar esta matriz es que la suma por renglón sea = 1. En la tabla 6. se muestra un extracto de la matriz de probabilidades de transición entre estados.

✓ $B = \{b_j(v_k)\}$ es la distribución de probabilidad de los sucesos observables representándose cada suceso por $w_1, w_2, \ldots w_M$ donde M es el total de elementos del diccionario de datos. Para obtener esta matriz, se obtienen los datos a partir del diccionario de datos. De igual manera se obtienen los unigramas y bigramas, así como sus frecuencias y probabilidades.

- Primero se obtienen los unigramas de las palabras (w_i) que conforman el diccionario y para la frecuencia de unigramas se realiza la suma de cada palabra (w_i) que aparece en el diccionario. Lo expresado anteriormente se muestra en la figura 2.
- De igual manera como se muestra en la figura 3, se obtienen los unigramas de cada estado (q_i) que aparece en el diccionario y sus respectivas frecuencias.

Tabla 6. Extracto de la matriz de probabilidades de transición entre estados formado del corpus CONLL.

	Nom	sig_pa	sig_pc	sig_com	num	sust
nom	0	0.068696	0.56957	0.12348	0	0
Sig_pa	0.32346	0	0	0	0.11945	0.45676
Sig_pc	0.008945	0.015206	0	0.3712	0.093918	0.02057
Sig_com	0.0053	0.0008834	0	0	0.057013	0.11199
num	0	0.0056054	0.024183	0.054773	0.24359	0.44299

$$unigramas = \text{palabra}(w_i) \text{que existe en el diccionario}$$

$$Frecuencia\ unigramas = \#\ \text{total de palabra}\ (w_i) \text{que existe en el diccionario}$$

$$i = 1,2,...,M$$

Fig. 2. Representación de obtención de unigramas de palabras o sucesos del diccionario de datos.

$$unigramas = cada\ estado(q_i) igual\ que\ aparece\ en\ el\ diccionario$$

$$frecuencia\ de\ unigramas = \#\ total\ de\ cada\ estado(qi) igual\ en\ el\ diccionario$$

Fig. 3. Representación de obtención de unigramas de etiquetas o estados del diccionario de datos.

- Para obtener los bigramas se toman pares formados por palabras – etiquetas del diccionario de datos, posteriormente se realiza el mismo proceso para obtener unigramas, la suma de cada par palabra – etiqueta, su frecuencia y probabilidades. Estos procesos se muestran en la figura 4.
- El tamaño de la matriz de probabilidades de los sucesos observables es de j estados por k sucesos observables del diccionario de datos. La condición para formar esta matriz es que la suma de cada j estados sea = 1. Un pequeño extracto que conforma esta matriz se muestra en la tabla 7.

Con el cálculo de todos los elementos de los modelos de Markov, se tienen los datos de entrada para la aplicación del algoritmo de Viterbi, junto con el texto de entrada que se definió anteriormente.

$$bigramas = cada\ par\ palabra - etiqueta\ (w_i, q_i) del\ diccionario\ de\ datos$$

$$frecuencia\ de\ bigramas = \#\ total\ de\ cada\ par\ palabra - etiqueta\ (w_i, q_i) del\ diccionario\ de\ datos$$

$$probabilidades\ de\ bigramas = \frac{frecuencia\ de\ cada\ par\ palabra - etiqueta(w_i, q_i) del\ corpus}{\#\ total\ de\ frecuencias\ de\ cada\ etiqueta(q_i)}$$

$$i = 1,2,....,M$$

Fig. 4. Representación de obtención de bigramas palabra-etiqueta del diccionario de datos.

$$b_j(v_k) = \textit{probabilidad de que la palabra}(w_k)\textit{tenga el estado}(q_j)$$

$$j = 1,2,....,N$$

$$k = 1,2,....,M$$

Fig. 5. Representación de distribución de probabilidad de los sucesos observables.

Tabla 7. Extracto de la matriz de probabilidades de los sucesos observables formado del corpus CONLL.

	General	judicial	básicos	popular	CrimeNet	web
sig_com	0	0	0	0	0	0
Num	0	0	0	0	0	0
Sust	8.92E-05	0	0	0	8.92E-05	0

2.2.4. Frecuencias relativas. Cuando se dispone de un texto etiquetado, la primera idea que acude a nuestra mente para abordar el proceso de estimación de los parámetros del modelo es quizás la de diseñar un sencillo mecanismo basado en el uso de frecuencias relativas [5]. Este es el otro proceso de etiquetado con modelos de Markov Ocultos que se realiza en esta investigación. El objetivo es determinar la secuencia de etiquetas óptima para una frase determinada de entre un conjunto de secuencias de etiquetas.

Para llevar a cabo este mecanismo, se obtienen unigramas y bigramas, así como sus frecuencias y probabilidades de las etiquetas o estados del cuerpo de entrenamiento, representándose Q como el total de elementos del corpus de entrenamiento.

2.3. Métricas de efectividad

Cualquier clasificador produce resultados, y por lo tanto, la ordenación de listas de categorías potenciales de etiquetas, o la toma de decisiones binarias para asignar categorías. Un método aplicable para la evaluación de un clasificador de resultados no puede aplicarse a un método binario [9]. Presentamos evaluaciones apropiadas para los dos casos e indicamos en las siguientes comparaciones cuales son usadas.

De acuerdo a Van Rijsbergen et al[8], las definiciones de precision y recall son las siguientes:

- ✓ Precision es la proporción del número de documentos relevantes recuperados del número total de documentos recuperados.
- ✓ Recall es la proporción del número de documentos relevantes recuperados del número total de documentos relevantes.

Markus Junker et al[6], define Accuracy y Error como sigue:

- ✓ Accuracy es definido como la proporción del número de documentos clasificados correctamente de entre el número total de documentos.
- ✓ Error es definido como 1-accuracy.

Yiming Yang et al [9], define F_B como sigue:

- ✓ La medida F_B es de uso frecuente como criterio de optimización en la formación de umbrales para las decisiones binarias.

2.3.1. Evaluación de clasificación de categorías. El recall y la precisión de una clasificación de categoría es similar a las medidas correspondientes usadas en la recuperación de texto. Considerando un documento como la entrada a un clasificador, y un listado de categorías ordenados como salida, el recall y la precisión en un umbral particular en esta lista ordenada son definidas como [9]:

$$recall = \frac{\text{categorias encontradas y correctas}}{\text{total de categorias correctas}} \qquad precision = \frac{\text{categorias encontradas y correctas}}{\text{total de categorias encontradas}}$$

donde categorías encontradas significa que las categorías están por encima del umbral. Para una colección de documentos de prueba, la clasificación de categorías para cada documento es evaluado en primer lugar, entonces el rendimiento de los resultados son promediados a través de los documentos. La precisión convencional en promedio es de 11 puntos el cual es usada para medir el funcionamiento de un clasificador sobre una colección de documentos [9].

2.3.2. Evaluación de clasificación binaria. Las medidas de ejecución en una clasificación binaria pueden ser definidas usando una tabla de contingencia de doble dirección (ver tabla 8). La tabla contiene cuatro celdas [9].

a = # de etiquetas correctamente asignadas

b = # de etiquetas incorrectamente asignadas

c = # de etiquetas correctas no asignadas

d = # de etiquetas incorrectas no asignadas

Tabla 8. Tabla de contingencia

	Etiquetas Correctas	Etiquetas incorrectas
Etiquetas asignadas correctamente	a	b
Etiquetas asignadas incorrectamente	c	d

El recall (r), precision (p), error (e) y accuracy (f) son definidas como:

$r = \frac{a}{(a+c)}$ *if* $a + c > 0$, *en otro caso* $r = 1$;

$p = \frac{a}{(a+b)}$ *if* $a + b > 0$, *en otro caso* $p = 1$;

$e = \frac{(b+c)}{n}$ *where* $n = a+b+c+d > 0$; $accuracy = \frac{(a+d)}{n}$ *where* $n = a+b+c+d > 0$;

Dado un clasificador, los valores de r, p, e y f a menudo dependen de los parámetros de ajuste interno; hay en general un equilibrio entre recall y precisión. Otra medida comúnmente usada es la llamada F-medida, que se define como:

$$F_{\beta} = \frac{(B^2+1)pr}{B^2 p + r}$$

donde B es el parámetro que da igual importancia a recall y precisión [9].

3 Resultados Experimentales.

Una vez finalizado el entrenamiento de los diferentes corpus y la ejecución de cada una de las técnicas así como de la arquitectura propuesta; se procede a la recopilación de las etiquetas que arrojaron las diferentes técnicas y las etiquetas predichas que conforman el 10% del corpus para prueba, estas etiquetas las sometemos a un análisis mediante un programa de evaluación utilizando las métricas de efectividad obteniendo los porcentajes de Recall, Precision, Accuracy, Error, y FB_1

3.1 Resultados con el Corpus CONLL

En la tabla 9 se muestran los resultados de cada técnica así como de la arquitectura propuesta utilizando el corpus CONLL y se observa que el Algoritmo de Viterbi es la técnica que mejor resultados tiene a nivel oración con un total de 297 oraciones etiquetadas correctamente, representado un 87.10%, de un total de 341 oraciones. Si bien en esta forma de evaluación al arquitectura propuesta no muestra buenos resultados, una vez más, hacemos hincapié, que el etiquetado que genera esta propuesta, es más confiable porque no sólo nos quedamos con el etiquetado de una sola técnica, sino que se reafirma o mejora con el etiquetado de otra técnica más.

Es importante mencionar que en este corpus, al momento de evaluar por oraciones con las diferentes técnicas, las mejoras que se registraron en cada una, fueron en su mayoría por las etiquetas sust (sustantivo), adj (adjetivo) y pron (pronombre), así como en las etiquetas asignadas incorrectamente en su mayoría fueron pron_rel (pronombre relativo) y adj (adjetivos), esto es debido a las altas probabilidades que tienen en comparación con las demás etiquetas.

Tabla 9. Resultados a nivel oración de todas las técnicas.

CONLL	Total de oraciones etiquetadas correctamente	Porcentaje
Chart Parser	233	68.33%
Viterbi	297	87.10%
Frecuencias Relativas	225	65.98%
Arquitectura propuesta	280	82.11%

En la tabla 10 se muestran los resultados de las métricas de efectividad para cada técnica utilizando el corpus CONLL. Si bien los resultados reflejados en la tabla 10, no favorecen a la arquitectura propuesta, no quiere decir que no sea eficiente, ya que es necesario mencionar que esta combinación de técnicas reafirman el etiquetado de una o más técnicas o lo mejoran.

Tabla 10. Resultados por oración aplicando las métricas de efectividad.

CONLL	Recall	Precision	Accuracy	Error	FB
Viterbi	98.65%	87.99%	87.10%	12.90%	93.02%
Frecuencias Relativas	99.10%	65.77%	64.81%	35.19%	79.06%
Chart Parser	100.00%	71.77%	70.09%	29.91%	83.57%
Arquitectura propuesta	99.28%	82.28%	80.94%	19.06%	89.98%

3.2 Resultados con el Corpus CLIC_TALP

En la tabla 11 se muestran los resultados de cada técnica así como de la arquitectura propuesta utilizando el corpus CLIC-TALP y se observa que una vez más el Algoritmo de Viterbi es la técnica que mejor resultados tiene a nivel oración con un total de 288 oraciones etiquetadas correctamente, representado un 82.52%, de un total de 349 oraciones.

En la tabla 12 se muestran los resultados de las métricas de efectividad, en donde observamos que el algoritmo de Viterbi tiene mejores resultados, si bien, la arquitectura propuesta no refleja buenos resultados, se puede decir que el etiquetado que genera es confiable, dado que emplea más de una técnica para reafirmar o mejorar el etiquetado, situación que no es considerada en la aplicación de cada algoritmo por separado, ya que en la arquitectura que se propone se da la oportunidad de etiquetar nuevamente una frase cuando existe la posibilidad de que las palabras tengan ambigüedad.

Tabla 11. Resultados a nivel oración de todas las técnicas.

CLIC-TALP	Total de oraciones etiquetadas correctamente	Porcentaje
Chart Parser	205	58.74%
Viterbi	288	82.52%
Frecuencias Relativas	274	78.51%
Arquitectura propuesta	224	64.18%

Tabla 12. Resultados por oración aplicando las métricas de efectividad

CLIC-TALP	Recall	Precision	Accuracy	Error	FB
Viterbi	98.95%	82.99%	81.95%	18.05%	90.27%
Frecuencias Relativas	99.63%	78.30%	76.79%	23.21%	87.68%
Chart Parser	98.64%	63.93%	63.32%	36.68%	77.58%
Arquitectura propuesta	98.65%	64.22%	63.61%	36.39%	77.80%

4 Conclusiones

La aplicación de las técnicas de etiquetado que hemos utilizado en la presente investigación han dado buenos resultados en diversos trabajos de investigación que existen actualmente, porque hacen uso de algún complemento, ya sea algún modelo simbólico, estadístico o ambos, para mejorar su eficiencia porque cada vez más es necesario obtener un etiquetado correcto y un reducido porcentaje de ambigüedad, siendo éste último uno de los principales puntos que actualmente se está tratando de combatir con las diferentes metodologías que existen para el Procesamiento del Lenguaje Natural escrito.

Los resultados obtenidos por la arquitectura propuesta, en ambos corpus, obtienen un tercer lugar, dado que realiza una combinación de algoritmos, el etiquetado resultante es aun más confiable y eficiente con respecto a las dos primeras etiquetas con buenos resultados , dado que por ejemplo en el analizador "Chart Parser" elige sólo la etiqueta que encuentra primero en el diccionario de datos sin contemplar las otras posibles etiquetas, evitando así, la posibilidad de tener un mejor etiquetado de acuerdo al contexto en la oración que se esté analizando en ese momento.

La diferencia entre tamaños de corpus (CONLL y CLIC-TALP) es un factor que influye en los resultados, porque entre más grande sea, habrá un mejor entrenamiento de éste y por lo tanto una mejor distribución de probabilidades que se reflejará en los resultados.

Referencias

1. A stochastic parts program and noun phrase parser for unrestricted text. In Proceedings of the Second Conference on Applied Natural Language Processing, pp. 136-143.

2. Civit T., Montserrat. "Guía para la anotación morfológica del corpus CLiC-TALP". Universidad Politécnica de Cataluña (2002) from www.lsi.upc.es/~nlp/tools/guia_anot.ps.gz.
3. Conference on Computational Natural Language Learning (CoNLL-2002). Last update: June 26, 2006. from http://www.cnts.ua.ac.be/conll2002/ner/data/
4. Rogelio Dávila. "Semantics and Parsing in Intuitionistic Categorial Grammar". PhD dissertation, University of Essex, June 1994.
5. Graña Gil, J., "Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural", Tesis Doctoral en Ciencias Computacionales. Universidad de Coruña, España.
6. Junker, M., Dengel A. and Hoch R. "On the evaluation of document analysis components by recall, precision and accuracy". German Research Center for Artificial Intelligence (DFKI) from http://citeseer.ist.psu.edu/cache/papers/cs/18899/http:zSzzSzwww.dfki.uni-kl.dezSz~junkerzSzdownloadzSzicdar99.pdf/junker99evaluation.pdf
7. Kim, S. Erik F. Tjong. "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition". In: *Proceedings of CoNLL-2002*, Taipei, Taiwan, 2002. from http://www.cnts.ua.ac.be/conll2002/pdf/15558tjo.pdf
8. Van Rijsbergen C.J. "Information Retrieval". Departament of Computing Science. University of Glasgow. pp. 6.
9. Yang, Y. (1999). "An evaluation of statistical approaches to text categorization. Journal of Information Retrieval". To appear. http://citeseer.ist.psu.edu/yang97evaluation.html