RNPS No. 2143
ISSN 2072-6260
Versión Digital

SERIE GRIS

REPORTE TÉCNICO
Minería de Datos

Estado del arte de la extracción de entidades nombradas

Lic. Cynthia Costales Llerandi,
Dr. C. José Hernández Palancar

RT_011 **marzo 2010**

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
Ciudad de La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2143
ISSN 2072-6260
Versión Digital

SERIE GRIS

REPORTE TÉCNICO
Minería de Datos

Estado del arte de la extracción de entidades nombradas

Lic. Cynthia Costales Llerandi,
Dr. C. José Hernández Palancar

RT_011 **marzo 2010**

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
Ciudad de La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Índice

Estado del arte de la extracción de entidades nombradas

Lic. Cynthia Costales Llerandi, Dr. C. José Hernández Palancar

Centro de Aplicaciones de Tecnología de Avanzada, 7a #21812 e/ 218 y 222, Siboney, Playa, Ciudad de La Habana, Cuba
ccostales@cenatav.co.cu

RT_011 CENATAV
Fecha del camera ready: 30 de octubre de 2009

Resumen: Obtener información relacionada con nombres de personas, lugares u organizaciones es un problema muy común en distintas áreas del Procesamiento del Lenguaje Natural (PLN) por lo que es muy importante poder extraer este tipo de elementos de un documento. Una de las características que afectan la realización de esta tarea es la ambigüedad semántica de los nombres propios y que además la cantidad de nombres propios que existen es grande y es imposible hacer uso de diccionarios solamente. La Extracción de Entidades Nombradas (EEN) es un subproblema de Extracción de Información e involucra el procesamiento de documentos estructurados y no estructurados. En este trabajo se verá cómo los primeros trabajos en este campo se enfocaron en combinar abundantes reglas hechas por humanos, palabras trigger, diccionarios, entre otros. Sin embargo estos métodos requieren expertos de dominio para construir estas reglas y el conjunto de palabras. Posteriormente los trabajos se han enfocado en el uso de técnicas de Aprendizaje Automático y más recientemente en sistemas híbridos, los cuales combinan técnicas vistas con anterioridad. También veremos cómo la tarea de EEN se resuelve generalmente como un proceso en dos pasos: reconocimiento de las palabras que componen a la EN y clasificación de las mismas, aunque algunos trabajos lo hacen en un solo paso. Realizamos un resumen de estas técnicas y los resultados de algunos sistemas desarrollados, así como otros aspectos críticos de EEN tales como características y métodos de evaluación.

Palabras clave: entidades nombradas, reconocimiento de entidades nombradas, extracción de entidades nombradas

Abstract: Information regarding names of persons, places or organizations is a very common problem in various areas of Natural Language Processing (NLP), so it is very important to obtain this type of elements from a document. A characteristic that affects the realization of this task is the semantic ambiguity of proper names, besides the amount of proper names is huge and it is impossible to make use of dictionaries only. Named Entity Extraction (NEE) is a subproblem of Information Extraction (IE) and involves the processing of structured and unstructured documents. This work will study the first papers in this field focused on combining human-made rules, trigger words, dictionaries, among others. However, these methods require domain experts to build these rules and the set of words. Later work has focused on using machine learning techniques and more recently in hybrid systems, which combine these techniques. We will also see how the task of NEE is usually a two step process: recognition of words that make up the EN, and classification. Some authors do these steps in a single process. We summarize these techniques and the results of some developed systems and also some aspects regarding to features used and the evaluation methods.

Keyword: Named Entity, Named Entity Recognition, Named Entity Extraction

1 Introducción

Desde hace ya varios años la cantidad de información disponible en distintos formatos y fuentes ha tenido un enorme crecimiento debido a, entre otras cosas, la aparición de distintas tecnologías, como Internet. Por tanto, es de gran importancia desarrollar herramientas que permitan administrar y realizar búsquedas de ciertos elementos en un documento.

Uno de los mayores problemas en el análisis del lenguaje natural es la presencia de palabras desconocidas, especialmente nombres. Como los nombres tienen un gran porcentaje en el texto, pueden ser la pieza más importante de información en un texto [1].

El término Entidad Nombrada nació en las Conferencias de Entendimiento de Mensajes (MUC por sus siglas en inglés) donde se buscaba promover y evaluar la investigación en el área de Extracción de Información. En las MUC se acuñó la siguiente definición:

Entidad Nombrada es una palabra o secuencias de palabras que se identifican como nombre de persona, organización o lugar [2].

Las ENs pueden consistir de cualquier tipo de palabra: adverbios, preposiciones, adjetivos, e incluso algunos verbos, pero la mayoría de las ENs están compuestas de sustantivos. Las ENs son sintagmas nominales [3].

La Extracción (o Reconocimiento) de Entidades Nombradas (EEN) es una tarea de la lingüística computacional en la cual se busca clasificar cada palabra de un documento en una determinada categoría de una lista de categorías definida, como por ejemplo persona, lugar, organización o fecha [4].

En las distintas áreas del Procesamiento del Lenguaje Natural (PLN), un problema común es obtener información relevante relacionada con nombres de personas, lugares u organizaciones, por lo cual se vuelve importante el poder extraer y distinguir este tipo de elementos de todo el conjunto de palabras que componen a un documento. Aún cuando algunos elementos son relativamente fáciles de identificar mediante el uso de patrones (por ejemplo fechas o datos numéricos) existen otros elementos, como personas, lugares u organizaciones, que presentan otras dificultades para ser identificados como pertenecientes a un tipo específico. El extraer y distinguir este tipo de elementos es el objetivo de la Extracción de Entidades Nombradas (EEN).

La EEN es un subproblema de Extracción de Información e involucra el procesamiento de documentos estructurados y no estructurados [5].

La EEN puede definirse como la tarea de identificar y clasificar en un documento textual expresiones que identifican instancias de conceptos relevantes para algún dominio de aplicación [6].

Para los humanos esta tarea es intuitivamente simple, no siendo así para las computadoras. Se podría pensar que las ENs se pueden clasificar utilizando diccionarios, pues la mayoría son nombres propios, pero esto es intratable pues el conjunto de todos los nombres es muy grande y sería muy costoso (computacionalmente) hacer uso de estos recursos solamente. Más aun, aunque todos los nombres propios estén registrados en un diccionario, no es fácil decidir su significado. La mayoría de los problemas en EEN radican en la existencia de ambigüedad semántica (de significado), es decir, un nombre propio tiene diferentes significados de acuerdo al contexto. Por ejemplo: ¿Cuándo “Mayo” es un nombre de persona? ¿Y cuándo el nombre de un mes? Por esto es necesario construir sistemas capaces de aprovechar las regularidades que cumplen las diferentes clases de entidades.

Debido en gran medida a que la formulación inicial de la EEN se concentraba en encontrar nombres de entidades disjuntos, un problema importante en el área es la extracción de ENs anidadas, la cual no ha sido suficientemente tratada. Los esfuerzos para abordar el problema de la EEN anidados han estado confinados principalmente al dominio biomédico. Con el objetivo de reutilizar técnicas bien conocidas y que han demostrado ser exitosas en el caso no anidado, los sistemas que tratan el caso de las ENs anidadas tratan el problema como una etapa separada de postprocesamiento del problema clásico o como la combinación de varias instancias de éste.

1.1 Aplicaciones de EEN

Esta tarea es útil para muchos problemas tales como:

- Traducción automática, donde es importante reconocer ENs porque generalmente estas permanecen sin traducción y etiquetarlas para que permanezcan intactas permite mantener coherencia en los textos [5], [7].
- Recuperación de información [5], [7], [8].
- Extracción de Información [8], [6].
- Resúmenes automáticos [5], [7].

Por ejemplo, la clave de un procesador de preguntas es identificar el punto de pregunta (quién, qué, cuándo, dónde, etc.), así que en muchos casos el punto de pregunta corresponde a una EN. En textos de biología, el sistema de EN puede extraer automáticamente los nombres predefinidos (como los nombres de proteínas) de documentos.

Un sistema de EEN puede usarse como el primer paso en una cadena de procesamientos: un próximo nivel de procesamiento podría relacionar 2 o más ENs, o tal vez incluso dar semántica a una relación usando un verbo. Un sistema de consulta en Internet podría usarlo para construir consultas formadas más apropiadamente: “¿Cuándo nació Bill Gates?” podría producir la consulta “Bill Gates”+nació.

También es aplicable al indexado automático de libros. Para muchos libros y documentos la mayor parte de los términos más informativos respecto a su contenido son ENs.

Una de las tareas de extracción de información es la de extracción de patrones de relaciones, donde el objetivo es encontrar la relación entre pares de entidades nombradas. Por ejemplo de la frase “Raúl Castro participará en los debates en la ONU”, se espera que un sistema con patrones de relaciones responda que “Raúl Castro” es una persona, “ONU” es una organización y que participará en un debate en la organización “ONU”. Pero antes de poder establecer esta relación, es necesario clasificar correctamente los elementos “Raúl Castro” y “ONU” como una persona y una organización respectivamente.

1.2 Evaluación

La evaluación de sistemas de Extracción de Información (EI) hace uso de las medidas de precisión, recall y de la medida F, las cuales han sido adoptadas como medidas estándar en el área de EEN.

En las evaluaciones MUC y MET, una respuesta correcta es aquella donde la etiqueta y los límites están correctos. Una respuesta está medio correcta si la etiqueta (el tipo y el atributo) está correcta pero solo 1 límite está correcto. Alternativamente, también es medio correcto si solo el tipo de la etiqueta (y no el atributo) y ambos límites están correctos [9].

Un modelo de puntuación desarrollado para las evaluaciones MUC y MET mide la precisión (P) y el *recall* (R) como se muestra en la Ecuación 1 y en la Ecuación 2:

$$\mathrm{P} = \frac{\#\text{ de respuestas correctas}}{\#\text{ de respuestas}}$$

Ecuación 1. Fórmula para medir la precisión

$$\mathrm{R} = \frac{\#\text{ de respuestas correctas}}{\#\text{ de correctas en llave}}$$

Ecuación 2. Fórmula para medir el *recall*

El término *respuesta* es usado para denotar "respuesta dada por el sistema", el término *llave* es usado para denotar "un fichero anotado que contiene las respuestas correctas". Informalmente, R mide el número de "hits" vs. el número de posibles respuestas correctas como se especifica en el fichero llave, mientras que P se define como una medida de la proporción de elementos clasificados por el sistema que en realidad son correctos [9].

Estas 2 medidas de rendimiento se combinan para formar una tercera medida, la medida F, la cual se calcula como se muestra en la Ecuación 3:

$$\mathrm{F}_{\beta} = \frac{(\beta^2 + 1)\mathrm{RP}}{(\beta^2\mathrm{R}) + \mathrm{P}}$$

Ecuación 3. Fórmula para calcular la medida F

Donde β es un factor que determina la importancia que se le da a cada una de estas medidas. Típicamente tiene valor 1:

$$\mathrm{F} = \frac{\mathrm{RP}}{1/2(\mathrm{R}+\mathrm{P})}$$

1.3 Foros de evaluación

MUC (Message Understanding Conferences) y MET (Multilingual Entity Task)

Desde el comienzo de la década de los 90, las conferencias o evaluaciones MUC (Message Understanding Conferences) se han consolidado en el desarrollo de métricas y algoritmos estadísticos para la realización de evaluaciones de sistemas basados en las tecnologías emergentes de la extracción de la información (EI). Estas conferencias han permitido la evaluación y comparación de diversos sistemas, siendo uno de los principales foros para la promoción de la extracción de la información [10, 11].

A mediados de la década de los 90, las evaluaciones MUC empezaron a suministrar datos y definiciones de tareas, además de proporcionar un software de evaluación totalmente automatizado de puntuación para medir el rendimiento de los sistemas de EI [11, 12].

Los resultados de estas evaluaciones fueron presentados en estas conferencias durante los años 90 en las que tanto desarrolladores como evaluadores pusieron en común sus logros y establecieron los futuros objetivos a afrontar en el desarrollo de sistemas de extracción de información.

En el año 1995 las conferencias MUC incluyeron una tarea de Reconocimiento de Entidades Nombradas [13], la cual determinó a lo que usualmente nos referimos con el término Entidad Nombrada, y estableció medidas estándares para la precisión de un sistema que realice esta tarea. Esta tarea fue introducida en el marco de la conferencia MUC-6. Inicialmente, el único idioma tratado fue el inglés, aunque posteriormente se desarrollaron versiones asiáticas de las conferencias MUC, en las cuales se trató el chino y el japonés.

Todos los participantes en la conferencia desarrollaron sistemas que realizaban tareas de comprensión del lenguaje natural definidas por el comité de la conferencia. Los sistemas eran evaluados basados en la forma en que su salida se comparaba con la salida de lingüistas humanos [14].

En MUC, esta tarea se divide en 3 subtareas:

1. Extracción de nombres (ENAMEX) (nombres de personas, lugares y organizaciones).
2. Extracción de expresiones temporales (TIMEX) (expresiones temporales (dates, times)).
3. Extracción de números (NUMEX) (cantidades monetarias y porcentajes).

En esta conferencia no se consideraba solapamiento ni anidamiento entre las ENs a reconocer.

Los conjuntos de datos utilizados en MUC-6 y MUC-7 son material privado propiedad del Linguistic Data Consortium (LDC) pero están disponibles para su venta. Los conjuntos de datos aplicados en las evaluaciones MUC-4, MUC-3 y MUC-2 están disponibles de forma gratuita en la web.

La evaluación MUC-6 de EEN demostró que los sistemas están acercando su rendimiento al humano en textos en inglés.

En MUC-6 15 organizaciones participaron en la evaluación de EN, incluyendo 2 que presentaron 2 configuraciones de los sistemas para las pruebas y uno que presentó 4, para un total de 20 sistemas. La medida F para la mitad de los sistemas probados fue de más de 90% [15].

El sistema de mejor rendimiento alcanzó una medida F de 96.42% (R=96, P=97), muy cercana al rendimiento humano. En esta evaluación 11 sistemas obtuvieron medidas F de más del 90%, y 5 de más del 85%.

Por su parte, en MUC-7 los idiomas tratados fueron el chino, el inglés y el japonés. Los resultados obtenidos por los sistemas participantes son los presentados en las tablas 1, 2 y 3, y en las figuras 1, 2 y 3 respectivamente.

- Idioma: chino

Tabla 1. Resultados de la evaluación de los sistemas en MUC-7 para el chino [16]

Sistema	P	R	F
Kent Ridge Digital Labs Official Summary Scores	**90**	**83**	**86.38**
Kent Ridge Digital Labs Un-Official Summary Scores	84	77	80.73
National Taiwan University	83	77	79.61

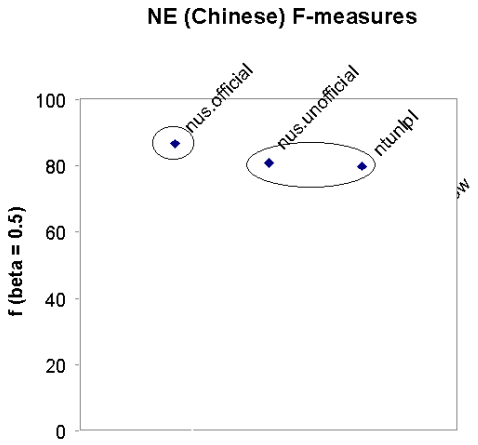

Fig. 1. Resultados de la medida F para el idioma chino [17]

- Idioma: Inglés

Tabla 2. Resultados de la evaluación de los sistemas en MUC-7 para el inglés [16]

Sistema	P	R	F
Annotator 1	98	98	97.60
Annotator 2	96	98	96.95
BBN	89	92	90.44
FACILE	78	87	81.91
IsoQuest system 1	90	93	91.60
IsoQuest system 2	74	93	82.61
Kent Ridge Digital Labs (NUS)	76	80	77.74
Language Technology Group	**92**	**95**	**93.39**
The MITRE Corporation	85	86	85.31
National Taiwan University	66	73	69.67
New York University	85	93	88.80
OKI	77	92	84.05
University of Durham	75	78	76.43
University of Manitoba system 1	85	87	86.37
University of Manitoba system 2	79	89	83.70
University of Sheffield	83	89	85.83

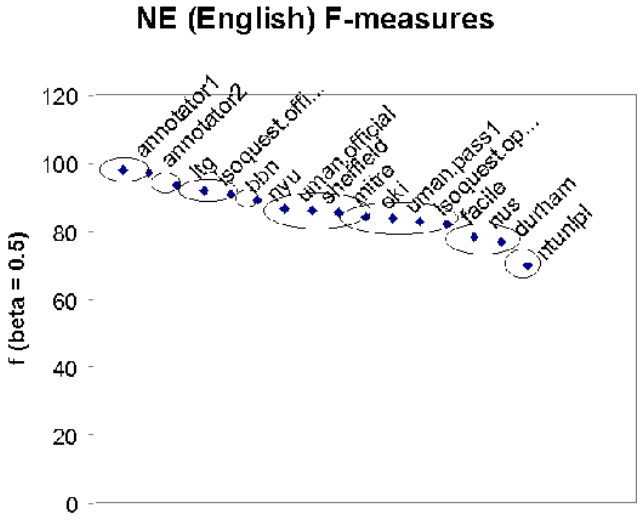

Fig. 2. Resultados de la medida F para el idioma inglés [18]

- Idioma: japonés

Tabla 3. Resultados de la evaluación de los sistemas en MUC-7 para el japonés [16]

Sistema	P	R	F
New York University	75	85	79.51
NTT	79	89	83.72
OKI	**85**	**97**	**90.54**

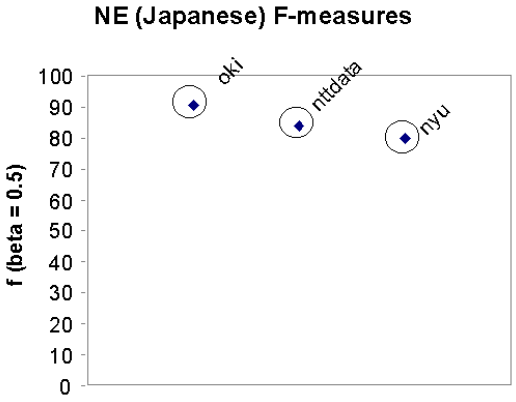

Fig. 3. Resultados de la medida F para el idioma japonés [19]

CoNLL

Las conferencias CoNLL se especializan en soluciones a problemas de PLN mediante métodos de aprendizaje automático. En el marco de éstas se realiza una competición llamada *Shared Task* (tarea compartida) en la que se evalúan los resultados de diferentes técnicas de aprendizaje en un

entorno experimental común. Las tareas compartidas del 2002 y 2003[1] trataron sobre la EEN independiente del lenguaje.

En la tarea compartida de 2002 se concentraron en 4 tipos de categorías: personas, lugares, organizaciones y misceláneas. A los participantes se les dieron datos de entrenamiento y prueba para al menos 2 lenguajes. Ellos usaron estos datos para desarrollar un sistema de EEN que incluyera un componente de aprendizaje de máquina. En esta competición pueden usarse fuentes de información aparte de las de los datos de entrenamiento. En la edición del 2002 los idiomas abordados fueron el español y el holandés [20].

En esta conferencia tampoco se tenía en cuenta el solapamiento ni anidamiento entre las entidades a reconocer.

En CoNLL-2002 participaron 12 sistemas, los cuales usaron una amplia variedad de técnicas de aprendizaje de máquina. Los resultados obtenidos son los mostrados en las tablas 4 y 5.

Tabla 4. Resultados obtenidos por los sistemas participantes en CoNLL-2002 para el idioma español

Sistema	P	R	F
Carreras et al, 2002a **[21]**	**81.38%**	**81.40%**	**81.39**
Florian, 2002 [22]	78.70%	79.40%	79.05
Cucerzan y Yarowsky, 2002 [23]	78.19%	76.14%	77.15
Wu et al, 2002 [24]	75.85%	77.38%	76.61
Burger et al, 2002 [25]	74.19%	77.44%	75.78
Sang, 2002 [26]	76.00%	75.55%	75.78
Patrick et al, 2002 [27]	74.32%	73.52%	73.92
Jansche, 2002 [28]	74.03%	73.76%	73.89
Malouf, 2002 [29]	73.93%	73.39%	73.66
Tsukamoto et al, 2002 [30]	69.04%	74.12%	71.49

Tabla 5. Resultados obtenidos por los sistemas participantes en CoNLL-2002 para el idioma holandés

Sistema	P	R	F
Carreras et al, 2002a **[21]**	**77.83%**	**76.29%**	**77.05**
Wu et al, 2002 [24]	76.95%	73.83%	75.36
Florian, 2002 [22]	75.10%	74.89%	74.99
Burger et al, 2002 [25]	72.69%	72.45%	72.57
Cucerzan y Yarowsky, 2002 [23]	73.03%	71.62%	72.31
Patrick et al, 2002 [27]	74.01%	68.90%	71.36
Sang, 2002 [26]	72.56%	68.88%	70.67
Jansche, 2002 [28]	70.11%	69.26%	69.68
Malouf, 2002 [29]	70.88%	65.50%	68.08
Tsukamoto et al, 2002 [30]	57.33%	65.02%	60.93

El sistema presentado po;r Carreras et al. [21] funcionó mejor que todos los demás sistemas por un margen significativo, tanto en español como en holandés. Debe tenerse en cuenta que ellos usaron alguna información adicional además de los datos de entrenamiento en el

[1] Ver http://www.cnts.ua.ac.be/conll2002/ner/ y http://www.cnts.ua.ac.be/conll2003/ner/

experimento en español. Sin esta información adicional su sistema (F=79.28) no se comporta significativamente mejor que el de Florian ([22]) (F=79.05).

En la tarea compartida de CoNLL-2003 participaron 16 sistemas. En esta edición del 2003 los idiomas abordados fueron el inglés y el alemán. Los resultados se muestran en las tablas 6 y 7.

Tabla 6. Resultados obtenidos por los sistemas participantes en CoNLL-2003 para el idioma inglés

Sistema	P	R	F
Florian et al, 2003 **[31]**	**88.99%**	**88.54%**	**88.76±0.7**
Chieu et al, 2003 [32]	88.12%	88.51%	88.31±0.7
Klein et al, 2003 [33]	85.93%	86.21%	86.07±0.8
Zhang et al, 2003 [34]	86.13%	84.88%	85.50±0.9
Carreras et al, 2003a [35]	84.05%	85.96%	85.00±0.8
Curran et al, 2003 [36]	84.29%	85.50%	84.89±0.9
Mayfield et al, 2003 [37]	84.45%	84.90%	84.67±1.0
Carreras et al, 2003b [38]	85.81%	82.84%	84.30±0.9
McCallum et al, 2003 [39]	84.52%	83.55%	84.04±0.9
Bender et al, 2003 [40]	84.68%	83.18%	83.92±1.0
Munro et al, 2003 [41]	80.87%	84.21%	82.50±1.0
Wu et al, 2003 [42]	82.02%	81.39%	81.70±0.9
Whitelaw et al, 2003 [43]	81.60%	78.05%	79.78±1.0

Tabla 7. Resultados obtenidos por los sistemas participantes en CoNLL-2003 para el idioma alemán

Sistema	P	R	F
Florian et al, 2003 **[31]**	**83.87%**	63.71%	**72.41±1.3**
Klein et al, 2003 [33]	80.38%	65.04%	71.90±1.2
Zhang et al, 2003 [34]	82.00%	63.03%	71.27±1.5
Mayfield et al, 2003 [37]	75.97%	64.82%	69.96±1.4
Carreras et al, 2003a [35]	75.47%	63.82%	69.15±1.3
Bender et al, 2003 [40]	74.82%	63.82%	68.88±1.3
Curran et al, 2003 [36]	75.61%	62.46%	68.41±1.4
McCallum et al, 2003 [39]	75.97%	61.72%	68.11±1.4
Munro et al, 2003 [41]	69.37%	**66.21%**	67.75±1.4
Carreras et al, 2003b [38]	77.83%	58.02%	66.48±1.5
Wu et al, 2003 [42]	75.20%	59.35%	66.34±1.3
Chieu et al, 2003 [32]	76.83%	57.34%	65.67±1.4
Whitelaw et al, 2003 [43]	71.05%	44.11%	54.43±1.4

Los resultados del sistema de Wu et al. ([42]) para los datos en inglés fueron corregidos después del deadline y la nueva medida F fue 82.69.

SemEval

SemEval (International Workshop on Semantic Annotations) es un foro para la evaluación de sistemas en tareas semánticas. La tarea 9 de SemEval 2007 (Anotación Semántica de múltiples

niveles del catalán y el español) se divide en 3 subtareas entre las que se encuentra la EEN. En esta subatrea las EENs pueden estar anidadas. Se distinguieron 6 categorías semánticas: Persona, Organización, Lugar, Fecha, Expresiones numéricas y Otros [44].

LREC

La Conferencia Internacional en Recursos del Lenguaje y Evaluación (LREC: siglas en Inglés) es organizada por ELRA cada 2 años con el apoyo de instituciones y organizaciones involucradas en HLT (*Human Language Technologies*).

1.4 Extracción de Entidades

La Extracción de Entidades Nombradas (EEN) consiste en clasificar cada palabra en un documento en algunas categorías predefinidas. En la taxonomía de tareas de lingüística computacional, esta cae en el dominio de "Extracción de Información".

La tarea de EEN se resuelve generalmente como un proceso en dos pasos, los cuales se describen a continuación:

Reconocimiento de Entidades Nombradas (REN): En este paso se determina la palabra o secuencia de palabras que componen a la EN. Este proceso en si es también una clasificación, donde el objetivo es clasificar cada palabra con una etiqueta de acuerdo a su pertenencia a una EN. La tarea de delimitación en la mayoría de las propuestas que atacan el problema de REN, se ve como un problema de clasificación de 3 clases (B, I, O), donde se utilizan generalmente características léxicas, sintácticas, ortográficas, afijos y características de los elementos alrededor de la palabra a clasificar [4].

En la delimitación de las ENs son de mayor importancia atributos como la aparición de letras mayúsculas y la ubicación de las palabras en la oración. También se ha utilizado información de los vecinos (también de tipo ortográfico) para apoyar la delimitación, la cual se ha resuelto bajo técnicas manuales, basadas en reglas o patrones, o bajo técnicas de aprendizaje automático que utilizan distintos atributos para entrenar clasificadores que aprendan los patrones de la delimitación de ENs.

Clasificación de Entidades Nombradas (CEN): Una vez que se han definido los límites de las ENs, se debe definir de qué tipo son, es decir, a qué clase pertenecen. Las clases definidas generalmente son: número, porciento, fecha, tiempo, persona, organización, lugar y miscelánea. En esta subtarea también se han utilizado distintas técnicas que implican técnicas "manuales", técnicas de aprendizaje automático y sistemas híbridos. Para la clasificación, los atributos a utilizar son similares a los utilizados en la primera fase pero pueden integrar algunos otros, como palabras "disparadoras", diccionarios, y las palabras de las ENs en sí. También se utilizan atributos de los vecinos que rodean a la EN. En el paso de clasificación, atributos como la posición de las palabras en la oración e información ortográfica son menos útiles para discriminar entre clases de ENs. Existe un gran número de casos ambiguos que hacen difícil alcanzar mayores niveles de desempeño en los sistemas de este tipo. Esta segunda fase de la EEN, la clasificación, es la que presenta mayores dificultades [4].

1.5 Tipos de entidades

En las conferencias MUC solo hay 3 tipos de etiquetas, cada una de las cuales usa un atributo para especificar una entidad particular. Los tipos de etiqueta y las entidades que denotan se definen como sigue:

i. Entidades (ENAMEX): persona, organización, lugar.
ii. Expresiones temporales (TIMEX): fecha, tiempo.
iii. Expresiones numéricas (NUMEX): cantidades monetarias, por ciento.

La mayoría de los sistemas de EEN clasifican de acuerdo a estas etiquetas.

Sin embargo, Sekine y Nobata ([45]) presentaron una jerarquía de ENs conteniendo 150 tipos de ENs y después presentaron otra con 200 ([46]).

2 Métodos existentes

En los últimos años, los sistemas de EEN se han convertido en una de las áreas de investigación más populares. Estos sistemas pueden categorizarse en 3 clases:

- Enfoques manuales (o basados en ingeniería del conocimiento)
- Basados en métodos de Aprendizaje Automático
- Híbridos

Los primeros trabajos de EEN se enfocaron en construir e integrar listas de diccionarios, de palabras disparadoras, de reglas o patrones y obtener recursos etiquetados a mano para poder hacer el reconocimiento de ENs con sistemas basados en reglas. Con el paso de los años, el uso de técnicas de aprendizaje automático se ha vuelto popular para la resolución de problemas propios del área de PLN y por supuesto en la tarea de EEN.

Los primeros enfoques usaron típicamente patrones de estado finito hechos a mano, los cuales intentaban machear contra una secuencia de palabras de la misma forma que lo hace un macheador de expresiones regulares [47].

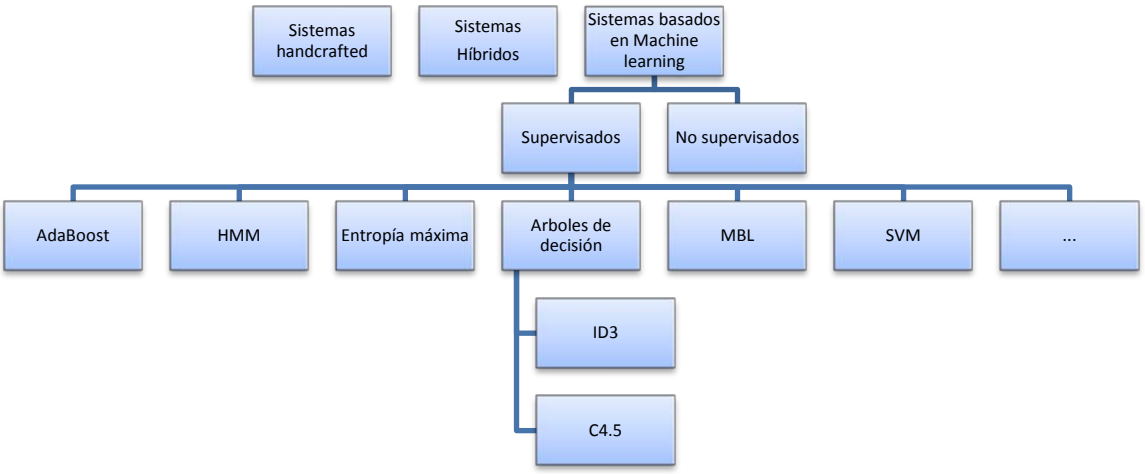

Fig. 4. Taxonomía de métodos para EEN

2.1 Sistemas Handcrafted

Los sistemas de este tipo son aquellos que son construidos a mano y que por tanto están basados en el conocimiento de los especialistas que los diseñan. Generalmente utilizan un conjunto de reglas y heurísticas que guían el proceso de etiquetado de los textos.

Los recursos más utilizados en este tipo de sistemas son las expresiones regulares, los conjuntos de reglas y los *gazetteers* (colecciones de nombres conocidos que se recopilan y procesan manualmente o de forma semiautomática).

Generalmente los sistemas consisten de un conjunto de patrones usando características gramaticales (por ej.: etiquetas POS), sintácticas (por ej.: precedencia de palabras) y ortográficas (por ej.: capitalización) en combinación con diccionarios.

Muchos de los sistemas que participaron en las tareas compartidas de MUC-6 usaron este enfoque.

Algunas palabras generalmente son parte de o están seguidas por (o precedidas) una EN particular. Sin embargo, esto no se cumple para todas las palabras pues para la mayoría, solo a veces ocurre esto. Ejemplos del primer caso son “Mr.”, “Prof.”, seguidos por un nombre propio. Ejemplo de lo otro son los nombres de compañías seguidos por “produce”. Los sistemas basados en reglas usan este fenómeno en 2 pasos: primero, las palabras que indican EN se recolectan. Segundo, las reglas son escritas tomando en cuenta la ocurrencia de otras palabras así como otras evidencia, tales como POS, la aparición de letras en minúsculas y mayúsculas, etc. Combinado con heurísticas adicionales, diccionarios geográficos u otras listas conteniendo nombres propios se obtienen buenos resultados, pero conlleva un gran trabajo.

Este tipo de sistemas tienen la ventaja de que pueden obtener muy buenos resultados, en su mayoría alcanzan porcentajes de precisión mayores al 90%. Este enfoque alcanza su mejor

rendimiento usando diferentes conjuntos de reglas construidas a mano por cada par lenguaje/dominio [48].

Estos tipos de modelos tienen mejores resultados para dominios restringidos, son capaces de detectar entidades complejas que representan un problema para los modelos basados en aprendizaje automático [5].

Sin embargo, este tipo de enfoque carece de la habilidad de hacer frente a los problemas de robustez y portabilidad. Cada nueva fuente de texto o idioma requiere una revisión significativa de las reglas para mantener un rendimiento óptimo y los costos de mantenimiento podrían ser bastante elevados. Esto significa que el conocimiento base completo de un sistema basado en reglas hechas a mano tiene que ser reescrito si el sistema se aplicara a un nuevo dominio o idioma. Además de la falta de portabilidad, construir una base de reglas de tamaño efectivo es muy costoso en términos de tiempo y dinero: NYU e IsoQuest reportaron gastar una persona por mes cada uno en la escritura de la base de reglas. Más aun, este enorme gasto se requiere cada vez que el sistema necesite ser portado a un nuevo dominio o lenguaje [3]. El costo de estos sistemas es elevado, ya que descansan en la experticia de lingüistas entrenados computacionalmente. Otra desventaja está en que el rendimiento será altamente sensible a la habilidad del lingüista computacional en la escritura de los patrones de EN y a la cantidad de labor dedicada a la tarea.

Antes de ser reconocida en 1995 la tarea de EEN, se realizaron sistemas que extraían nombres propios de textos. Un trabajo publicado en 1991 por Lisa F. Rau [1] es citado como la raíz del campo. En este trabajo se provee un método para extraer nombres de compañías de un texto. Este método usa una combinación de heurísticas, listas de excepciones, y análisis del corpus. El método localiza un indicador de nombre de compañía (por ej.: CO, NV, SA, INC) y lee hacia atrás a partir de este sufijo un máximo de 6 palabras, sin incluir los signos de puntuación para determinar dónde comienza el nombre de la compañía. Si no ocurre ninguna condición de parada, las 6 palabras se toman como constituyentes del nombre de compañía y se extraen. Estas condiciones de parada son listas de palabras que en caso de ocurrir alguna en el texto, indican el comienzo de la EN [1].

Por su parte, Appelt et al. participaron en la evaluación MUC-6 con un sistema de EEN basado en expresiones regulares construidas manualmente llamado FASTUS. Ellos dividen la tarea en 3 pasos: reconocimiento de frases, reconocimiento de patrones y resuelven incidentes. Fastus es uno de los sistemas más robustos sin embargo su mantenimiento es trabajoso [49].

Otro de los sistemas presentados en MUC-6 de este tipo fue el “Proteus”, el cual fue presentado por NYU. En este sistema, compuesto por un gran número de reglas de reducción sensibles al contexto, un nombre es reconocido como de cierto tipo si está definido en un diccionario, si tiene una forma distintiva (un patrón definido) o si es un alias de un nombre de tipo conocido. Por ejemplo, una regla es “*Título Palabra_Mayuscula => Título Nombre_Persona*”, la cual está bien para muchos casos, pero para “Mrs. Field’s Cookies” está mal pues este es el nombre de una compañía [50].

El sistema de EEN presentado por Black, Rinaldi y Mowatt era un módulo de un sistema mayor llamado FACILE. El módulo de EN de FACILE no empleó ningún método de aprendizaje, era completamente basado en reglas y utiliza bases de datos de ENs comunes. Se asignaron pesos explícitos a las reglas para resolver los conflictos entre distintas reglas que predecían distintas clases para una EN [51]. Debajo hay una regla de ejemplo de este modulo que busca nombres de universidades y las clasifica como organizaciones: “si una ciudad, región

o ciudad conocida está precedida por la expresión "*university of*", esa expresión completa debe marcarse como una organización con un factor de confianza de 0.9":

[syn = NP, sem = ORG] (0.9) => [norm = "Universidad"], [token = "de"], [sem = REGION|COUNTRY|CITY];

Un sistema muy similar a Facile es IsoQuest [52], el cual también se basa en reglas escritas manualmente y utilizaban bases de datos de ENs comunes.

Los autores de este trabajo hicieron uso de listas de nombres en su sistema. Ellos descubrieron que reducir su tamaño por más del 90% tenía poca repercusión en el rendimiento, por el contrario, añadir solo 42 entradas llevó a mejorar los resultados. Esto implica que la calidad de la lista es más importante que el número total de entradas en la efectividad.

En el sistema LOLITA hacen uso de una gran base de conocimiento (una red semántica), la cual es un hipergrafo dirigido de 100,000 nodos que mantiene información tal como jerarquías de conceptos e información léxica. El algoritmo trabaja examinando los conceptos creados en la red semántica. La gran mayoría de la red (70%) proviene de WordNet. Cada una de las etapas de procesamiento del sistema está implementada basada en reglas [53].

2.2 Métodos basados en aprendizaje automático

El aprendizaje automático es una rama de la Inteligencia Artificial cuyo objetivo es desarrollar técnicas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos.

La posibilidad de no necesitar especialistas para examinar los datos para encontrar reglas fue una fuerte motivación para comenzar las investigaciones en algoritmos de aprendizaje automático [54].

Actualmente, la tendencia en la EEN es utilizar métodos de aprendizaje automático debido a que pueden adaptarse con mayor facilidad a diferentes dominios. No obstante, posibles errores en el etiquetado manual de los corpus, así como las limitaciones propias del método de aprendizaje utilizado, pueden afectar la calidad de los resultados obtenidos por estos sistemas.

En la práctica, los sistemas basados en aprendizaje supervisado han obtenido mejores resultados que aquellos basados en aprendizaje no supervisado. No obstante, la disponibilidad enormemente superior de textos no etiquetados ha hecho que se preste atención además a los métodos semisupervisados.

2.2.1 Métodos supervisados

Para utilizar algoritmos de aprendizaje supervisado es necesario contar con un conjunto de textos (denominado corpus) etiquetado con los nombres de entidades que sirvan como ejemplos para el entrenamiento. La tarea de etiquetar este corpus también requiere un esfuerzo considerable, sin embargo, el grado de especialización del personal dedicado a ella no necesita ser tan alto como el del personal que define los conjuntos de reglas en los sistemas manuales.

2.2.1.1 Árboles de decisión

Bennett et al. construyen varios árboles de decisión para la tarea de EEN. Por cada clase de entidad construyen 2 árboles de decisión binarios, uno para predecir la palabra de comienzo y

otro para la de cierre usando el algoritmo C4.5. El sistema creado fue nombrado RoboTag y fue aplicado en los idiomas inglés y japonés [48].

Uno de los factores que llevó a los autores a utilizar árboles de decisión fue que el procedimiento de etiquetado inducido por el sistema pudiera ser fácilmente explicado en términos de cómo realiza las decisiones (para lograr la confianza del usuario) y otros algoritmos de aprendizaje o estadísticos (tales como redes neuronales o HMM) no ofrecían esta ventaja.

Para extraer las características de aprendizaje se emplea un preprocesador para cada lenguaje con el que se opera. Este preprocesador realiza tokenización, segmentación de palabra, análisis morfológico, y *lookup* léxico. La salida producida por este preprocesador es utilizada por el algoritmo de aprendizaje representándola como vectores de características. Para la clasificación también se tienen en cuenta características de los tokens alrededor del token a analizar.

Al sistema hay que especificarle varios parámetros que afectan su rendimiento como por ejemplo el número de tokens usado para hacer cada tupla de entrenamiento. Estos parámetros deben ser ajustados cada vez que se quiere aplicar el sistema en un nuevo dominio, lo cual es una dificultad del sistema. En las pruebas realizadas con distintos corpus los valores de los parámetros con los cuales se obtuvo los mejores resultados fueron diferentes.

Cuando se etiqueta un texto se hace uso de los árboles de decisión aprendidos para producir una lista de clases potenciales de comienzo y fin por cada tipo de clase, es decir, el sistema realiza múltiples decisiones en cada token, por lo que se le pueden asignar múltiples clases posiblemente inconsistentes. Existen muchas formas de emparejar estas clases de comienzo y fin, por lo que el algoritmo de macheo debe decidir el mejor emparejamiento de éstas. Ellos resolvieron el problema introduciendo 2 métodos: uno de ellos es la medida de distancia, la cual es usada para encontrar un par de inicio y fin para cada EN basado principalmente en la información de distancia y el otro es el esquema de prioridad de etiquetas, que escoge una EN entre diferentes tipos de candidatos que se solapan basado en el orden de prioridad de las ENs.

Los autores realizaron pruebas para el inglés con el corpus de *Wall Street Journal* de MUC-6 y para el japonés con un corpus de MET. Las reglas para las fechas y números fueron hechas a mano y haciendo uso de estas y de los árboles de decisión para el resto de las clases, la medida F en la tarea de MUC-6 para inglés es 90.1%.

Una variante del trabajo de Bennett et al. [48] fue usada en el sistema presentado por la Universidad de *New York* (NYU) en MET-2 para el idioma japonés presentado en los trabajos [55] y [56].

El sistema presentado en estos 2 últimos trabajos no trabaja completamente automático, pero se desempeña bien con un pequeño corpus de entrenamiento y no tiene parámetros para ajustar manualmente, lo cual es una ventaja respecto al sistema de Bennett et al. Otra diferencia con este sistema es que solo se construye un árbol de decisión con el algoritmo C4.5. Usan 3 tipos de conjuntos de características en el árbol de decisión: POS, tipo de carácter (Kanji, Katakana, número o símbolo, etc.) y diccionarios especiales. En la fase de entrenamiento se construye un árbol de decisión que aprende sobre los inicios y finales de las ENs basado en los 3 tipos de información de los tokens previo, actual y siguiente. El árbol de decisión da una salida para cada token.

Si solo se hace uso de la decisión determinista creada por el árbol, podría haber un problema en la fase de corrida porque las decisiones son hechas localmente, por lo que el sistema podría realizar una secuencia inconsistente de decisiones (Por ejemplo, un token podría etiquetarse como el inicio de una organización, y el próximo token como el cierre del nombre de una persona). Para resolver esto los autores usan un método probabilístico. Una vez asignadas todas

las probabilidades a todos los tokens en una oración, la tarea es descubrir el camino consistente más probable en la oración. La salida es generada de la secuencia consistente con la mayor probabilidad par cada oración. El algoritmo Viterbi se utiliza en esta búsqueda.

La creación de diccionarios se realizaba a mano, lo cual se podría automatizar usando un método de *bootstrapping* con el cual comenzando con diccionarios base, se puede correr el sistema en textos no etiquetados e incrementar las entidades en los diccionarios.

Por su parte, el enfoque de Paliouras et al. [57] se parece al usado en los sistemas analizados anteriormente ya que hace uso del algoritmo C4.5. La principal diferencia de este trabajo con los previamente publicados que hacen uso de este algoritmo para EEN, está en la representación del problema. Estos enfoques tratan de identificar los componentes de una frase perteneciente a un tipo particular de EN, especialmente sus puntos de comienzo y fin, por lo que requieren de una etapa más de post-procesamiento en la que se construye una frase basada en sus componentes individuales.

La mejor solución presentada a ese problema es buscar la secuencia más probable de etiquetas que provee una solución válida. Una objeción a esto es que introduce conocimiento que es externo al árbol de decisión inducido. Como resultado, el árbol de decisión – y el conjunto de reglas asociado al cual puede traducirse - cesa de ser de uso directo del experto humano, pues no puede ser usado por sí mismo para identificar ENs. De esta forma, el enfoque de árboles de decisión pierde su ventaja de ser directamente interpretable por los humanos.

En contraste con este enfoque de post-procesamiento, proponen un paso pre-procesamiento, en el cual los sintagmas nominales son identificados por un parser. Bajo la asunción débil de que las ENs son sintagmas nominales, el árbol de decisión puede entonces enfocarse en estas frases y clasificarlas en los tipos de ENs requeridas.

Este sistema hace uso de un conjunto de listas de gazetteers, consistentes en nombres de personas, organizaciones, designadores de compañías (Ltd., Co.), títulos de personas (Mr.), etc., y una gramática. La información tomada en cuenta por la gramática consiste en etiquetas asignadas revisando las listas de gazetteers, PoS y propiedades sintácticas de las palabras en una frase. Un parser usa esta gramática para identificar frases de interés en el texto. La adaptación de tal sistema a un dominio particular involucra la actualización de las listas de gazetteer y la gramática. En este trabajo se simplifica el problema de la adaptación considerando solo la gramática. Las listas gazetteer necesitan ser construidas de antemano.

Cada instancia de EN es representada por un vector de características. Las características utilizadas son la etiqueta de gazetteer, si tiene, y el PoS. Se tiene en cuenta además las 2 palabras adyacentes a cada lado de la frase.

El árbol de decisión generado realiza ambas partes de la tarea EEN, es decir, la identificación y clasificación de las frases EN.

En el experimento diferentes niveles de poda del árbol se examinaron, induciendo árboles de decisión de varios tamaños. Para cada tamaño se realizaron pruebas para estimar el rendimiento del sistema y al parecer no hay fluctuación significativa en los resultados al aumentar el tamaño de los árboles, pues los árboles pequeños y simples se desempeñan tan bien como árboles más grandes.

Los resultados obtenidos son comparables con los presentados por RoboTag.

En el trabajo presentado por Sánchez se enfocan en la subtarea de clasificación, la cual considera que es la que más problemas presenta. El trabajo propuesto consiste en un método para aprovechar la información de las distintas menciones de una misma EN refinando una clasificación inicial, para lo cual se presenta un proceso en cuatro pasos, los cuales involucran las dos fases clásicas de la EEN (delimitación (REN) y clasificación inicial (CEN)) y otras dos

fases que son las que se proponen en este trabajo: la vinculación de ENs y el refinamiento de la clasificación inicial [4].

En la primera etapa de la solución propuesta, la vinculación de ENs, se calcula para cada una de las ENs la similitud que presentan con el resto de las ENs. Para esto se experimentó con cuatro medidas de similitud (Similitud exacta, Similitud Dice, Similitud de Superposición, Contención exacta de una cadena en otra). Se decidió utilizar el método de similitud exacta de una cadena en otra pues brinda mayor precisión, lo cual es necesario porque los resultados de la segunda etapa se verán influenciados por el resultado de la primera etapa.

En la segunda etapa (paso de Clasificación Final) se buscó corregir los errores que se tienen en la clasificación inicial, haciendo uso de la información proporcionada por el paso anterior. Se realizaron un conjunto de pruebas con los métodos de voto simple y voto ponderado y los resultados obtenidos fueron que con la votación ponderada los resultados son un poco más altos que en el caso del voto simple, sin embargo no se logran mejoras sustanciales con respecto a la clasificación inicial. Por esta razón utilizó otro enfoque basado en árboles de decisión. Se construyeron árboles para generar reglas para cada una de las clases (MISC, ORG, LOC y PER).

También se realizaron pruebas para determinar el número de elementos de la EN a considerar y se determinó que tomar un solo elemento antecesor y sucesor eran suficientes, ya que al tener en cuenta más elementos en la ventana los resultados de clasificación se obtenían resultados menores.

Como resultado obtuvieron que los resultados obtenidos con el uso de los métodos presentados en este trabajo dependen de la naturaleza de los datos con los que se trabaja (por ejemplo: el tamaño de los documentos y el dominio de los mismos). Además influyen el método de clasificación inicial, los errores de delimitación generados en el primer paso y los errores de la clasificación inicial. La integración de la información de las vinculaciones de las ENs bajo los distintos enfoques propuestos no representó una mejora notable con respecto a la clasificación inicial, en los casos en los que se mejoraron los valores fue de forma mínima.

2.2.1.2 HMM

El problema de EEN puede ser visto como un problema de clasificación, donde se clasifica cada palabra como perteneciente a una de las clases de EN o no. Una de las técnicas más populares para tratar con la clasificación de secuencias es Modelos Ocultos de Markov (HMM por sus siglas en inglés (Hidden Markov Model)) [58]. Los HMMs son autómatas de estado finito con transiciones entre estados y emisiones de símbolos.

La filosofía del funcionamiento de un HMM es la siguiente. Se escoge un estado inicial aleatoriamente de acuerdo con la función de probabilidad inicial y dicho estado emite una observación de acuerdo con una función de probabilidad de emisión. Se escoge un nuevo estado aleatoriamente de acuerdo con la probabilidad de transición y en este nuevo estado se emite nuevamente una observación. Estos últimos dos pasos se repiten de modo que se genera una secuencia de observaciones. Para buscar una solución globalmente óptima de acuerdo a los presupuestos del modelo, se emplea el algoritmo de Viterbi.

Una variante del problema de aprendizaje consiste en ajustar las probabilidades de inicio, transición y emisión a las de un conjunto de secuencias de observaciones de las que se conoce las secuencias de estados por las que debe pasar el modelo para generarlas. En este caso, cada una de esas probabilidades puede ser calculada directamente de los datos de entrenamiento ya

que no es necesario buscar en el espacio de todas las distribuciones de probabilidades de transición. Los modelos se entrenan según esta variante del problema de aprendizaje y la fase de etiquetado se trata como una instancia del problema de la decodificación, o sea, dada una secuencia de palabras (o frases), se determina la secuencia de estados más probable por la que pudo pasar el modelo para generarla y se mapea dicha secuencia al conjunto de clases utilizado. Ejemplos de estos enfoques son Nymble [9] y SIFT [59], ambos desarrollados para el idioma inglés.

El sistema presentando por Bikel et al., llamado Nymble (posteriormente llamado IdentiFinder, el cual es un producto comercial muy conocido y muy efectivo), está basado en Modelo Ocultos de Markov (HMM por sus siglas en inglés (Hidden Harkov Models)) y fue el primer sistema de EEN estadístico de alto rendimiento [9].

Para los autores la EEN puede ser vista como un problema de clasificación. Debido al éxito de los HMMs en otros problemas de clasificación de textos, escogieron desarrollar una variante de un HMM para esta tarea.

Este modelo asigna a cada palabra uno de los tipos de EN o la etiqueta NOT-A-NAME (para representar las palabras que no pertenecen a ninguna EN). Los autores organizaron estos estados en regiones como se puede ver en la Figura 5. Dentro de cada una de estas regiones utilizan un modelo del lenguaje bigram donde cada palabra está representada por un estado, y hay una probabilidad asociada con cada transición de la palabra actual a la próxima palabra.

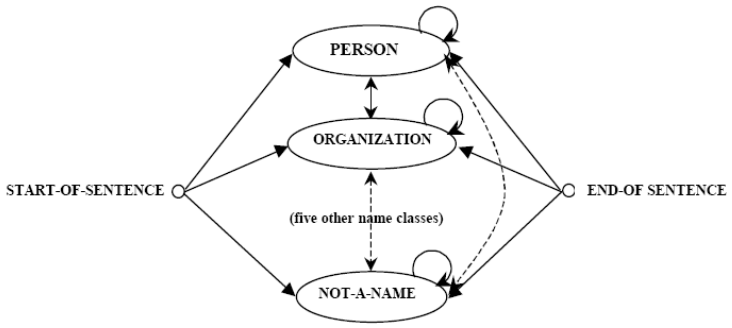

Fig. 5: Representación pictórica del modelo conceptual

El trabajo del modelo generativo es encontrar la secuencia más probable de clases (NC) dada una secuencia de palabras (W):

$$\max \mathrm{P_r}(\mathrm{NC}|\mathrm{W}) = \max \frac{\Pr(\mathrm{W}, \mathrm{NC})}{\Pr(\mathrm{W})}$$

Haciendo uso del algoritmo Viterbi buscan el espacio completo de todas las posibles asignaciones de tipos de ENs, maximizando el numerador de la ecuación (el denominador es constante para cualquier secuencia de palabras (W)). El uso de este algoritmo permite encontrar la secuencia óptima de estados eficientemente (lo hace en tiempo lineal en el número de tokens en una oración).

Las palabras son consideradas como pares ordenados compuestos por una palabra y una característica de palabra, denotado por $< w, f >$. La característica de palabra es un cálculo realizado a cada palabra el cual produce 1 de 14 valores predeterminados y que son calculados en un orden determinado para evitar ambigüedad. Esta es la parte del modelo que es dependiente del lenguaje.

Las características usadas en la versión de MUC-7 del sistema incluyeron muchas características pertenecientes a expresiones numéricas, capitalización, y pertenencia a listas de palabras importantes.

Los autores hicieron varios experimentos para obtener la cantidad óptima de palabras de entrenamiento. Como resultado obtuvieron que usando cerca de 100,000 palabras en el entrenamiento se logra un rendimiento comparable con sistemas hechos a mano.

Para MUC-7, BBN presentó el sistema SIFT el cual realizaba, entre otras tareas, la de EEN. Algunos de los autores de este trabajo son autores de Nymble. Para la tarea de EEN usan el sistema IdentiFinder [9], [59].

En este caso los autores también midieron el efecto del tamaño del corpus de entrenamiento en el rendimiento del sistema. Como resultado obtuvieron que con el incremento de la cantidad de datos de entrenamiento se logra una mejora de los resultados del sistema pues la medida F del sistema aumentó en 2 puntos con el incremento del tamaño de los datos de entrenamiento de 91,000 a 176,000 palabras. Sin embargo, aumentar más la cantidad de datos de entrenamiento solo logró un punto de incremento de esta medida, por lo que se obtiene como resultado que añadir más datos de entrenamiento tendrá progresivamente un efecto menor en el rendimiento del sistema.

El sistema IdentiFinder de BBN obtuvo un score del 94% en la tarea de EEN.

El sistema de Rössler está basado en Modelos de Markov de segundo orden[2] y no hace uso de gazetteers, solo de una lista de palabras, las cuales son extraídas mediante métodos estadísticos de un corpus no anotado [8]. En los 2 sistemas anteriormente analizados solo se tenía en cuenta 1 estado anterior.

La entrada del sistema consiste de un texto etiquetado con POS, excepto las palabras que ocurren en la lista obtenida. Con esta entrada pretendemos que el modelo aprenda alguno de los patrones especificados de las etiquetas POS en y alrededor de una EN.

Las palabras que proveen evidencia interna y externa al modelo se tomaron como aquellas que ocurren alrededor de una etiqueta en una ventana de 2 palabras antes y 2 después de una EN. Estas palabras se almacenaron en la lista con otras informaciones. Para reducir el ruido en la lista usaron una técnica de pesado TF*IDF (la cual es muy común en la recuperación de información) combinada con una medida que describe la probabilidad de ocurrir en una posición particular y una categoría de EN particular. Esta medida sirve para distinguir una palabra que, por ejemplo, siempre ocurre inmediatamente después de una categoría de EN particular de otra palabra que ocurre equitativamente distribuida en diferentes posiciones y cerca de diferentes categorías.

Usaron un filtro “aprender - aplicar - olvidar” el cual guarda todas las ENs encontradas en un artículo, el cual es leído nuevamente y aquellas ENs que no habían sido encontradas anteriormente son marcadas si han sido identificadas en otras posiciones en el mismo artículo. Después que un artículo es procesado, el sistema “olvida” las ENs aprendidas.

Comparando los resultados obtenidos con otros sistemas, los valores son bajos en general. Esto puede deberse a que el sistema no hace uso de gazetteers y trabaja en el contexto limitado de trigramas. El uso de gazetteers parece inevitable pues en los resultados mostrados por muchos autores se evidenció el efecto positivo de usar información de este tipo [8].

[2] Los Modelos de Markov son autómatas de estado finito con transiciones de estados probabilísticas y emisión de símbolos. Están definidos por las probabilidades de las transiciones de estado y la probabilidad de emitir un símbolo de salida por un estado particular. Modelos de Markov de segundo orden significa que las probabilidades de transición son calculadas considerando 2 estados anteriores en lugar de 1.

El enfoque detrás del sistema de Zhou et al. está basado en el etiquetador de *chunks* basado en HMM que fue el mejor sistema individual en CoNLL'2000. Aquí, una EN es estimada como un *chunk*, llamado "NE-Chunk" [47].

La evaluación de este sistema en las tareas de MUC-6 y MUC-7 en inglés logró una medida F de 96.6% y 94.1% respectivamente, los cuales son resultados muy buenos.

La premisa básica de este modelo es considerar el texto en bruto encontrado cuando se decodifica, como si hubiera pasado por un canal ruidoso, donde ha sido marcado originalmente con las etiquetas de las EN. La tarea del modelo es generar directamente las etiquetas de EN originales desde las palabras de salida del canal con ruido. Este modelo generativo es inverso al modelo generativo del HMM tradicional como se usa en IdentiFinder, el cual modela el proceso original que genera las palabras anotadas con la clase de EN. Otra diferencia es que el modelo de de Zhou et al. asume independencia mutua de información mientras que el HMM tradicional asume independencia de probabilidad. Esta asunción de de Zhou et al. es mucho más holgada que la otra. En este trabajo también hacen uso del algoritmo Viterbi [47].

En este trabajo los tokens se denotan también como pares ordenados de palabras y características. Este modelo captura 3 tipos de características internas: en la primera se considera información relevante para discriminar entre fechas, por cientos, tiempo y cantidades monetarias, y también información de capitalización; la segunda característica de este tipo es sobre palabras disparadoras que los autores consideran útiles para el reconocimiento de EN; y la ultima característica interna contiene información de gazetteers.

La evidencia externa se refiere al contexto de otras ENs ya reconocidas. En este sistema solo tienen 1 característica externa la que consiste en una lista que es actualizada con cada EN que se ha reconocido. Cuando un nuevo candidato de En se encuentra, un algoritmo de alias se invoca para determinar su relación con las ENs en la lista reconocida.

Según Curran y Clark, el relativamente bajo rendimiento de los sistemas usados en CoNLL-2002 fue mayormente debido a los conjuntos de características usados más que al método de aprendizaje, lo cual es evidenciado por el buen rendimiento del sistema de Zhou [47], los cuales usan gran variedad de características [36].

2.2.1.3 Entropía máxima

El sistema presentado por NYU en la tarea compartida de MUC-7 se llamó MENE [60], [3]. Este es un sistema híbrido que utiliza entropía máxima para combinar las salidas de varios sistemas basados en patrones hechos a mano y obtiene resultados superiores a los de cada uno de dichos sistemas independientemente, demostrando que estos métodos de aprendizaje proveen un alto grado de portabilidad a varios dominios y lenguajes.

La modelación estadística está dirigida al problema de construir un modelo estocástico para predecir el comportamiento de un proceso aleatorio. En la construcción de este modelo, típicamente se tiene una muestra de salida del proceso. Podemos entonces usar esta representación para realizar predicciones del futuro comportamiento del proceso [61].

Muchos problemas en el Procesamiento del Lenguaje Natural (PLN) pueden ser reformulados como problemas de clasificación estadísticos, en los cuales la tarea es estimar la probabilidad de que una clase a ocurra en un "contexto" b, $p(a, b)$. Los contextos en las tareas de PLN usualmente incluyen palabras, y el contexto exacto depende de la naturaleza de la tarea; para algunas tareas, el contexto b puede consistir en una palabra, mientras que para otras, b puede consistir de muchas palabras y sus etiquetas sintácticas asociadas. Los corpus de texto

usualmente contienen alguna información sobre la coocurrencia de as y bs, pero nunca la suficiente para especificar completamente $p(a,b)$ para todos los posibles pares (a,b), ya que las palabras en b son típicamente *sparse*. El problema entonces es encontrar un método para usar la evidencia *sparse* sobre as y bs para estimar con confianza un modelo de probabilidad $p(a,b)$ [62].

La entropía máxima es un método muy flexible de modelación estadística que hace uso de las nociones de "futuro", "historia" y "característica" [3]. Los "futuros" se definen como las posibles salidas del modelo. Una "historia" son todos los datos condicionantes que permiten asignar probabilidades al espacio de "futuros".

Una solución de entropía máxima al problema de EEN permite el cálculo de $p(f|h)$ para cualquier f del espacio de posible futuros F, para cada h del espacio de posibles historias H. En el problema de EN, podemos reformular esto en términos de encontrar la probabilidad de f asociada con el token en el índice t en el corpus de prueba como:

$$p(f|h_t) = p(f|\text{informacion derivable del corpus de prueba relativa al token t})$$

El cálculo de $p(f|h)$ en EM es dependiente de un conjunto de "características". Los autores de estos trabajos se restringen a características que son funciones binarias de la "historia" y "futuro" [3].

Dado un conjunto de características y algunos datos de entrenamiento, el proceso de estimación de EM produce un modelo en el cual cada característica g_i tiene asociado un parámetro α_i. Esto permite calcular la probabilidad condicional como sigue:

$$P(f|h) = \frac{\prod_i \alpha_i^{g_i(h,f)}}{Z_\alpha(h)}$$

$$Z_\alpha(h) = \sum_f \prod_i \alpha_i^{g_i(h,f)}$$

Según los autores este método permite al modelador concentrarse en seleccionar las características que mejor caracterizan el problema mientras dejan al estimador de EM ocuparse sobre asignarle a las características sus pesos relativos [3].

El sistema presentado por NYU en la tarea compartida de MUC-7 se llamó MENE [60], [3]. Este es un sistema híbrido que utiliza entropía máxima para combinar las salidas de varios sistemas basados en patrones hechos a mano y obtiene resultados superiores a los que obtiene cada uno de dichos sistemas trabajando de forma independiente, demostrando que estos métodos de aprendizaje proveen un alto grado de portabilidad a varios dominios y lenguajes.

2.2.1.4 Reglas de asociación

En el trabajo de Budi y Bressan se propone un método para el reconocimiento de EN basado reglas de asociación (RA). Las RAs que definen los patrones de sintaxis y características de los términos son minadas de un conjunto de documentos de entrenamiento [63].

Una RA tradicional es una relación de la forma: X ⇒ Y, donde X y Y son conjuntos de ítems del conjunto de datos a estudiar. A cada RA se le asigna un factor de soporte y un factor de confianza. El soporte es la proporción del número de elementos en X y Y sobre el número total

de elementos; y la confianza es la proporción del número de elementos en X y Y sobre el número de elementos en X. La minería de RAs consiste en extraer del conjunto de datos todas estas reglas con soporte y confianza mayores o iguales a un soporte o confianza especificados por el usuario.

En la tarea de EEN los conjuntos de datos son documentos (son secuencias de términos con características y nombres de clases) y los ítems son las ocurrencias de términos. Los autores usan el conjunto de características propuestas por Bikel et al. en Nymble [9] y consideran los 7 tipos de EN consideradas en MUC-7.

En este trabajo, luego de pruebas realizadas, se consideran los siguientes 3 tipos de reglas:

$< t_2 > \Rightarrow nc_2$ (soporte, confianza) reglas de diccionario

$< t_1, t_2 > \Rightarrow nc_2$ (soporte, confianza) reglas bigram

$< t_1, f_2 > \Rightarrow nc_2$ (soporte, confianza) reglas de característica

Donde $< t_1, t_2 >$ es una secuencia de términos, f_2 es la característica de t_2 y nc_2 es el tipo de EN de t_2.

En la fase de entrenamiento se construyen por cada EN encontrada en el texto estas 3 reglas, cada una con un soporte y confianza dependientes del número de ocurrencias de los términos y las clasificaciones dadas.

Una vez obtenidas las reglas minadas, se consideran para la tarea de EEN si su soporte y la confianza son mayores que ciertos umbrales definidos por el usuario. Estas reglas minadas son usadas por tipo –diccionario, bigram, o característica- independientemente o combinadas. Por cada par de términos en el texto, el algoritmo de reconocimiento de ENs determina la regla con soporte mínimo y mayor confianza a usar.

El trabajo fue evaluado en el corpus de MUC-7. Se realizaron varios experimentos combinando los tipos de reglas. Los mejores resultados en MUC-7 se obtuvieron con bigram+diccionarios o características+diccionarios.

Con este trabajo se presentó un nuevo método de reconocimiento de EN basado en RAs. Los experimentos mostraron que estas reglas consistentemente tienen una mayor precisión que el método de ME.

Este trabajo es parte de una propuesta mayor para el desarrollo de sistemas, herramientas y técnicas de recuperación de información y lingüístico. El primer paso de este proyecto consiste en identificar ENs en textos, pero consideramos que algunos aspectos del mismo pudieran ser analizados con mayor profundidad y mejorar el sistema:

- Las características usadas son "fijas", y son las mismas que utilizan en Nymble. Se deberían tener en cuenta otras características que pueden mejorar los resultados del sistema.
- Solo usan 1 palabra antes de la que están analizando, se podría usar una ventana más grande.
- Hacen uso de 3 tipos de Reglas de Asociación fijas. Se podrían realizar experimentos con más tipos de reglas, o no fijarlas de antemano.

2.2.1.5 Aprendizaje basado en transformaciones (TBL)

Black and Vasilakopoulos compararon el aprendizaje de árboles de decisión con TBL y encontraron que, a pesar de que aprendizaje mediante árboles de decisión se comportó adecuadamente, el TBL fue mejor en la tarea de EEN.

TBL es una técnica de Aprendizaje Automático que es un tanto similar a la inducción de los árboles de decisión pues el resultado final de este último puede verse como un conjunto de reglas que gobierna el proceso de clasificación; deduciendo este conjunto de reglas de transformación es el objetivo explícito de TBL [64].

Para entrenar un sistema basado en este tipo de aprendizaje se obtiene un clasificador inicial a partir de un subconjunto del conjunto de entrenamiento, asignando a cada palabra o frase la etiqueta más probable. Tomando como base este clasificador inicial, se etiqueta una muestra de referencia para obtener la lista de los errores que se comenten de acuerdo con el modelo inicial y, a partir de esta lista de errores, se induce el conjunto de reglas, así como el orden en que deben ser aplicadas. El proceso se repite iterativamente hasta que se satisfaga alguna condición de convergencia.

En el paper de Florian se presenta un enfoque de clasificador basado en *stacking* a la tarea de EEN. TBL, Snow y un algoritmo *forward-backward* son *stacked* (la salida de un clasificador es pasado como entrada al próximo clasificador), procurando una mejora considerable en el rendimiento [22].

Otro sistema que hace uso de este tipo de algoritmos es el de Milidiú et al. [65].

2.2.1.6 SVM

Según Cruz las máquinas de soporte vectorial (SVM, del inglés Support Vector Machine) son clasificadores lineales que inducen hiperplanos separadores siguiendo una estrategia denominada maximización del margen, la cual consiste en escoger, entre todos los (potencialmente infinitos) hiperplanos que separan los vectores pertenecientes a clases distintas, aquel cuyas distancias a los vectores más próximos de cada clase sea máxima [6]. Estos vectores, los cuales determinan la frontera de cada clase, se denominan vectores soporte.

Intuitivamente, este hiperplano logra la mejor generalización, ya que ocupa la posición más neutra posible respecto a los ejemplos de cada una de las clases. La Figura 6 ilustra la situación en $\mathbb{R}^2$.

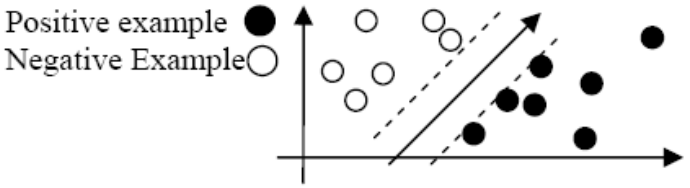

Fig. 6: Clasificación lineal con SVM

En las SVM, los objetos se representan como vectores de rasgos y la clasificación es binaria, por lo que habitualmente se utilizan tantos clasificadores como clases tenga el problema que se trata. Las salidas de estos clasificadores son posteriormente sometidas a algún criterio de evaluación que determine el resultado final.

En el método propuesto por Mansouri et al. se utiliza SVM y se aplica un algoritmo fuzzy para mejorar la clasificación del método SVM. Mediante esto se elimina el punto débil de los SVM en multiclasificación, ya que en métodos de clasificación normal cada EN pertenece a una clase fija basada en sus características. El primer paso en este sistema es segmentar los datos de

entrada en tokens y luego se seleccionan características tales como información léxica (unigram y bigram), afijos (2-4 letras sufijos y prefijos), información de la EN anterior, entre otras [5].

Para los datos de entrenamiento, el sistema hace uso de una clasificación lineal SVM y asigna una etiqueta a cada EN. Después de eso, cuando el sistema acepta el conjunto de datos de entrenamiento, se aplica un SVM fuzzy a los datos de prueba y de entrenamiento para reconocer la clase de EN exacta para cada token del conjunto de datos de prueba. De esta forma el sistema reconoce la clase dinámica para cada nombre con respecto al concepto de nombre en el texto o el documento completo. La función de pertenencia fuzzy que se produce en este método ayuda a reconocer nombres más semánticamente que los métodos existentes.

Un método supervisado que trata de ser lo más independiente posible del idioma es el descrito por McNamee et al. Lo que se usa de un idioma es una lista de las 1,000 palabras más frecuentes de dicho idioma. El método construye un clasificador para cada posible tipo de salida y aunque no obtiene muy buenos resultados (una F del 60 %), hay que tener en cuenta que usa pocos recursos lingüísticos [66].

Un resultado obtenido por Mayfield et al. fue que SVM es útil para un sistema de EEN cuando el número de características es grande [37].

2.2.1.7 Aprendizaje basado en memoria

El aprendizaje basado en memoria (MBL, del inglés *Memory Based Learning*) consiste en la extracción de un conjunto de ejemplos del corpus de entrenamiento. El algoritmo de los *K* vecinos más cercanos (k-NN) es un algoritmo de aprendizaje supervisado, el cual es un caso particular de MBL, donde el resultado de una consulta de una instancia nueva es clasificado basado en la categoría de la mayoría de los *K* vecinos más cercanos. Este algoritmo trabaja basado en la distancia de la consulta a los ejemplos de entrenamiento para determinar los K vecinos más cercanos.

Cruz afirma que uno de los motivos por el cual se ha empleado el MBL para la EEN, principalmente en el entorno académico, es que los esfuerzos de desarrollo se reducen debido a la disponibilidad de librerías de software libre como el TiMBL, que contienen implementaciones de este método fácilmente integrables con otro software [6].

El sistema presentado por Sang para la tarea compartida de CoNLL-2002 estuvo basado en MBL. Los autores usan 3 técnicas adicionales para mejorar el rendimiento del algoritmo de aprendizaje: *cascading*, selección de características y combinación de sistemas [26].

Se hace uso del algoritmo k-NN como clasificador básico. El algoritmo guarda todos los datos de entrenamiento y clasifica los nuevos datos comparándolos con los datos de entrenamiento. Los nuevos datos recibirán la misma clasificación que los datos de entrenamiento más similares a él. El paquete de software utilizado fue el TiMBL.

El conjunto inicial de parámetros utilizados para que el algoritmo prediga la mejor clase de entidad para una palabra consiste de la palabra y un grupo de palabras anteriores y posteriores. Para saber si las palabras del contexto forman parte de la EN también usan *cascading*, alimentando con la salida de un *learner* la entrada de otro.

Los autores buscaron el mejor conjunto de características automáticamente realizando "selección de características". Al haber muchos conjuntos diferentes de características para realizar una búsqueda completa hicieron uso de un método de búsqueda llamado *bi-directional hill-climbing* para explorar el espacio de características.

El rendimiento del sistema obtenido no fue tan bueno como los sistemas existentes para el inglés, sin embargo, se comporta razonablemente bien para el español y el holandés.

La fortaleza de este sistema es su habilidad de operar sin muchas pistas lingüísticas sobre el lenguaje procesado. Un texto con las entidades anotadas es suficiente para obtener un rendimiento aceptable. Una debilidad práctica es su velocidad de procesamiento: solo procesa 6 palabras por segundo en una máquina paralela.

En el trabajo de Hendrickx et al. se delimitan y etiquetan las ENs en un solo paso. Los autores entrenaron y probaron el sistema con los datos de la tarea compartida de CoNLL-2003. Primero, se incorporan características que señalan la presencia de wordforms en listas externas específicas del lenguaje. Segundo, construyeron un clasificador que corrige los errores de la salida de la primera etapa y por último, se añaden las instancias seleccionadas de los datos no anotados clasificados al material de entrenamiento. El sistema alcanza una medida F de 78.20 para el inglés y 63.02 para el alemán. En este trabajo la clasificación también se realiza haciendo uso del algoritmo k-NN [67].

2.2.1.8 AdaBoost

Al combinar clasificadores la idea es que al coincidir en la clase predicha, esta clasificación probablemente sea más correcta que la que da cada clasificador por separado.

Existen métodos de entrenamiento destinados a crear combinaciones de clasificadores simples que en conjunto obtengan un buen funcionamiento. AdaBoost es un algoritmo de este tipo.

Todos los sistemas analizados para representar el contexto local de una palabra w usan una ventana W centrada en la palabra. Este contexto es usado por un clasificador para hacer una decisión en la palabra. En la ventana cada palabra alrededor de w es codificada con un conjunto de características, junto con su posición relativa a w [24], [21], [7], [68].

En estos trabajos no tienen en cuenta el solapamiento de entidades.

El sistema presentado por Wu et al. fue diseñado para CoNLL-2002, por lo que fue probado con los idiomas español y holandés. Según estos autores, este es el primer intento de usar boosting para resolver el problema de la EEN, aunque el sistema descrito por Carreras et al. ([21]) también fue presentado en CoNLL 2002 y también usa AdaBoost [24].

El sistema expuesto por Carreras et al. en el 2003 es una réplica de otro sistema que fue expuesto en un trabajo anterior de los mismos autores con algunos cambios, el cual fue presentado para CoNLL-2003, por lo que los idiomas son el inglés y el alemán [68].

El sistema presentado por Wu et al. utiliza el algoritmo AdaBoost y los clasificadores débiles usados fueron los Decision Stump, que básicamente son un árbol de decisión de un solo nivel. El algoritmo que realmente usaron en este sistema es el AdaBoost.MH, el cual es una generalización del algoritmo AdaBoost para clasificación multiclase pues el algoritmo AdaBoost original fue diseñado para problemas de clasificación binarios, lo cual para los autores no cumple con los requerimientos de la tarea de EEN. Los autores parten de la idea que la tarea de EEN tiene propiedades similares a Categorización de Textos y por tanto, como se ha mostrado que AdaBoost.MH se desenvuelve bien en este problema, asumieron la hipótesis de que también se comportará bien en la EEN [24].

El sistema que proponen se basa en seleccionar un número de características fácilmente obtenibles para cualquier lenguaje. Estas características se usan para entrenar los clasificadores

débiles. Las características usadas en los experimentos finales fueron las palabras y lemas, etiquetas PoS, capitalización, entre otras.

Durante los experimentos realizados, los autores compilaron un diccionario a partir de la colección de entrenamiento con todas las ENs con más de 3 palabras y que tenían un tipo consistente (es decir, solo tenían 1 etiqueta de EN), pues se demostró que existía un número de ENs multi-palabra que no fueron identificadas correctamente, aparentemente debido a que el tamaño de la entidad fue un problema para el tamaño de la ventana de contexto. Adicionalmente, añadieron a la lista ENs correspondientes a nombres de clubes profesionales de deportes y países.

En los otros 3 sistemas las 2 principales subtareas del problema de la EEN (REN y CEN) son realizadas secuencial e independientemente con módulos separados. En ambos módulos hacen uso de clasificadores AdaBoost, combinando pequeños árboles de decisión de profundidad fija como reglas base [21], [7], [68]. Carreras et al. siguen el enfoque de realizar cada tarea tan pronto como la información para ello esté disponible, por ello el REN se realiza durante el análisis morfológico pues solo requiere información contextual en Word forms, patrones de capitalización, etc. La CEN se realiza después del análisis morfológico y antes del etiquetamiento porque no obtuvieron mejoras significativas cuando se añadió información de lema y POS [7].

Carreras et al. [21] usan AdaBoost binario con *confidence rated predictions* como algoritmo de aprendizaje para los clasificadores. Ese mismo año, estos autores usaron AdaBoost binario para ambas subtareas [7]. En el año 2003 utilizaron la versión binaria de AdaBoost en el módulo de reconocimiento y la extensión AdaBoost.MH en el módulo de clasificación [68].

En el primer trabajo mencionado los autores hicieron pruebas con las siguientes 3 variantes de esquemas de decisión para la subtarea de REN [21]:

- BIO: en este modelo se usan 3 clasificadores binarios, cada uno correspondiente a cada etiqueta. Las oraciones se etiquetan de izquierda a derecha, seleccionando para cada palabra la etiqueta con mayor confianza que es coherente con la solución actual (por ejemplo, Las etiquetas O no pueden ser seguidas por etiquetas I). Los 3 clasificadores usan la ventana para representar el contexto. Todas las palabras en el conjunto de entrenamiento son usadas como ejemplos de entrenamiento, aplicando una binarización one-vs-all[3].
- Open-Close&I: en este esquema se detecta la palabra que abre y la que cierra la EN. Las oraciones se procesan de izquierda a derecha, aplicando glotonamente 3 clasificadores ('open', 'close' y el clasificador I del esquema BIO). El clasificador 'open' representa el contexto con una ventana de la cual se extraen las características mientras que el 'close' hace uso de 2 ventanas.
- Open-Close Global: busca las palabras que abren y las que cierran la EN, pero realiza inferencia global para producir el conjunto de ENs, por lo que se usan 2 clasificadores.

El mejor resultado obtenido en este trabajo realizando REN con estos 3 esquemas fue con el BIO.

En otros trabajos Carreras et al. dicen que la forma más simple y eficiente de usar los clasificadores consiste en explorar la secuencia de palabras en una cierta dirección y aplicar los

[3] Se entrenan N clasificadores binarios y se entrenan para distinguir los ejemplos en una sola clase de los ejemplos de las demás clases. Cuando se desea clasificar un nuevo ejemplo, se corren los N clasificadores y el clasificador que de cómo resultado el mayor valor es el escogido. 69. Rifkin, R. and A. Klautau, *In Defense of One-Vs-All Classification.* 2004.

clasificadores coherentemente [7], [35]. Esto es un enfoque glotón. Esto es O(n), donde n es el número de palabras en la oración. Otra posibilidad es usar programación dinámica para asignar la secuencia de etiquetas que maximiza un score global sobre la secuencia de palabras. En ambos trabajos, por razones de simplicidad y eficiencias, se usó el enfoque glotón.

En el sistema presentado en el año 2002 también hicieron pruebas con los esquemas 'BIO' y Open-Close [7]. El primero se comportó mejor que el segundo. Sin embargo, al realizar pruebas con el esquema Open-Close&I, obtuvieron que este es el que mejor rendimiento obtuvo.

En un trabajo del año 2003 [35] usan solo el esquema BIO, por lo que usan 3 clasificadores. Todas las palabras en el conjunto de entrenamiento son utilizadas como ejemplos de entrenamiento, aplicando también una binarización one-vs-all. Estos autores afirman que a pesar de su simplicidad, este esquema se comporta muy bien y que otras representaciones de etiquetado más sofisticadas, estudiados en años anteriores [21], no mejoraron el rendimiento.

Los autores de estos 3 artículos ven la tarea de CEN como una tarea de clasificación, caso en el cual las decisiones son tomadas independientemente y la clasificación de una EN no influye en la de otra.

En el caso de un trabajo presentado en el 2002 [21] consiste en una tarea de clasificación de 4 clases consistente en asignar un tipo de EN a cada EN ya reconocida. Se realizaron muchas pruebas con matrices de combinación ECOC y como resultado obtuvieron que el mejor setting es el que combina todos los clasificadores. Binarizaron el problema entrenando hasta 10 clasificadores binarios diferentes: los 4 posibles clasificadores one-vs-all, las 3 posibles combinaciones (no simétricas) de clasificadores 2-vs-2, y 3 clasificadores binarios entrenados para distinguir entre 2 categorías ignorando las 2 restantes (PER-vs-LOC, PER-vs-ORG, LOC-vs-ORG). No usaron otras combinaciones para mantener el costo computacional en niveles adecuados. Estos clasificadores binarios son combinados usando Error Correcting Output Code (ECOC) con un esquema de decodificación basado en pérdida, lo que es, tomar en cuenta el grado de confianza de cada clasificador en lugar de simplemente su salida binaria.

En otro trabajo de ese mismo año la binarización del problema CEN consistió de un clasificador binario para cada clase, donde cada ocurrencia de entrenamiento es usada como un ejemplo positivo para su clase y como un ejemplo negativo para las restantes (esquema one-vs-all). La combinación de decisiones binarias se realiza seleccionando las clases a las cuales los predictores binarios asignaron un grado de confianza positivo. Todos los clasificadores usan un mismo conjunto de características [7].

Una de las conclusiones alcanzadas por los autores con este trabajo fue que en la CEN la combinación de los 4 clasificadores binarios obtiene menor rendimiento que cualquiera de ellos, por lo que otros esquemas de combinación deben explorarse, así como el uso de algoritmos AdaBoost multiclase.

En el año 2003 modelaron esta tarea como un problema de clasificación tanto de 3 clases (PER, ORG y LOC, en el cual MISC es una clase más) como de 4 clases (MISC, PER, ORG y LOC), lo cual resultó ser la mejor opción, por lo cual fue el que usaron. Según los autores la razón para usar AdaBoost.MH en vez de ECOC es que aunque este último provee resultados ligeramente mejores, su costo computacional es más alto que el de AdaBoost.MH [35].

Entre las características aplicadas a cada palabra en la ventana se encuentran características ortográficas y semánticas como la word form, bolsa de palabras y palabras disparadoras, entre otras. Una lista externa se usa para determinar si una palabra puede disparar (trigger) una cierta clase de EN y características gazetteer [21].

En otro trabajo usaron características del contexto y características de conocimiento externo. Para las de este último tipo usaron una lista de trigger words con personas, organizaciones, lugares, etc., y un gazetteer cuyas entradas contenían nombres de personas y geográficos. Estas listas fueron extraídas semiautomáticamente de la colección y recursos léxicos [7].

En la fase experimental, estos autores construyeron automáticamente una lista de palabras funcionales con el conjunto de entrenamiento para cada lenguaje con el cual probaron los sistemas [21], [35]. Similarmente, construyeron un gazetteer con las ENs en el conjunto de entrenamiento. En [21] además usaron conocimiento externo, dígase una lista de palabras trigger para ENs y un gazetteer externo. Estos recursos externos fueron reutilizados por [35].

El uso de fuentes de conocimiento adicionales tales como gazetteers externos y listas de palabras trigger les reportó una mejora de un 2% en el sistema CEN [21].

Se realizaron pruebas con varias combinaciones de características. Como resultado se obtuvo que el uso de información extra mejoró el rendimiento. El mejor resultado se logró cuando se usaron ambos recursos externos (gazetteer y lista de palabras trigger), dejando claro que cada uno de ellos provee información no incluida en el otro. En la tarea REN todas las características usadas son de contexto. Observaron empíricamente que la adición de conocimiento de gazetteers y trigger words provee solo evidencia muy débil para decidir la segmentación correcta de una EN [7].

Por cada problema de clasificación binaria entrenaron clasificadores con profundidades en el rango de 1 a 5 y con hasta 2,000 árboles base por clasificador. Refinaron la profundidad y el número de árboles a combinar optimizando el F_1 en el conjunto de desarrollo [21].

En otro trabajo entrenaron el sistema usando diferentes conjuntos de características y número de rondas de aprendizaje, usando clasificadores base de diferentes complejidades (desde decision stumps hasta árboles de decisión de profundidad 4). Como resultado obtuvieron que los árboles de decisión se comportaron mejor que los decision stumps en ambas subtareas y que aumentar aún más la profundidad de los árboles provee solo una pequeña ganancia [7].

Se hizo un preprocesamiento de filtrado de atributos para evitar el overfitting y para agilizar el aprendizaje. Para cada problema de clasificación se entrenaron los correspondientes clasificadores AdaBoost, aprendiendo hasta 4,000 árboles de decisión base por clasificador, con profundidades en el rango de 1 a 4. La profundidad de las reglas base y el número de rondas fueron optimizadas directamente en el conjunto de desarrollo [35].

Otro resultado importante obtenido por estos autores es que el rendimiento se degrada con el length de la secuencia a detectar, pero se obtuvieron buenos resultados para ENs de hasta 6 palabras [7].

En 3 de los trabajos analizados los autores evaluaron el rendimiento del módulo CEN con la salida de un sistema REN perfecto y obtuvieron mejores resultados, por tanto, el encadenamiento de los 2 módulos causa la propagación de errores y la degradación del rendimiento [21], [7], [35].

Un intento fallido realizado para mejorar el rendimiento consistió en etiquetar todo el conjunto de datos en 2 pasos: en el primero las ENs reconocidas se guardaron en un gazetteer temporal, usado en el segundo paso para producir el etiquetamiento final. La idea detrás de esto fue obtener ventaja de las repeticiones de una EN particular en el texto, pero no se logró ninguna mejora [21].

Una línea de mejora reportada podría ser añadir procesamiento dependiente del lenguaje al sistema. Una línea de investigación abierta es el enfoque simultáneo de REN y CEN, para que cada decisión pueda tomar ventaja de la sinergia entre ambos niveles de conocimiento [21].

Otro resultado obtenido es que en este enfoque, el conjunto de ENs que empiezan con palabras en minúscula es un problema para el módulo REN, especialmente debido al pobre tratamiento semántico de los ejemplos de entrenamiento. Los resultados son más bajos porque en muchas ocasiones el clasificador 'open' no tiene suficiente evidencia para comenzar una EN en minúscula [7].

Los autores plantearon que una mejora bajo investigación era probar otros algoritmos de clasificación diferentes a AdaBoost (aunque esta tarea se adapta bien a las capacidades de este algoritmo, otros algoritmos pueden ofrecer rendimientos similares o mejores (como Máquinas de Soporte Vectorial (SVM))) [7]. Otra de las mejoras bajo investigación plantadas fue que aunque los resultados iniciales no reportaron buenos resultados, creen que el uso de esquemas de inferencia global (en vez del enfoque glotón usado) para asignar la secuencia de etiquetas merece una mayor investigación. Esta misma idea de usar un procedimiento de etiquetado no glotón tendría chance de mejorar los resultados fue propuesta en [35].

2.2.2 Métodos semisupervisados

El etiquetado de un corpus es una tarea costosa, por lo cual no siempre se puede contar con la disponibilidad de suficientes muestras para el entrenamiento de los clasificadores. Sin embargo, existen enormes volúmenes de textos sin etiquetar a los que se puede acceder de forma fácil y económica. El objetivo del aprendizaje semi-supervisado es combinar muestras etiquetadas y muestras no etiquetadas para mejorar los clasificadores. La principal técnica en esta categoría es llamada "*bootstrapping*" [70]. En este método, un pequeño grado de supervisión, tal como un conjunto de semillas, se usa al principio. Por ejemplo, para extraer nombres de enfermedades, 5 nombres de enfermedades iniciales se proveen al sistema. Entonces el sistema busca las oraciones que contienen esos nombres, y encuentra indicadores fuertes de contexto donde los 5 nombres de enfermedades aparecen. Después el sistema trata de encontrar otras instancias que aparezcan en el contexto. Repitiendo estos procesos, se puede recuperar un gran número de nombres de enfermedades.

Algunos sistemas de este tipo explotan la redundancia entre las características internas y contextuales de las palabras [71], [72]. Un ejemplo de una pista interna de palabra es el hecho de que una palabra capitalizada comenzando con "Sr." es probable que pertenezca a la clase PERSON. Un ejemplo de una pista contextual es que una palabra seguida por "S.A." es probable que sea de tipo ORGANIZACION.

En el trabajo de Collins y Singer solo se trató la sub-tarea de clasificación de ENs (CEN). Estos autores introducen el algoritmo CoBoost para realizar la clasificación de EN, el cual hace uso de clasificadores que usan la ortografía de una EN o el contexto en el cual la entidad ocurre [72].

En este trabajo, los autores parsean un corpus completo en la búsqueda de patrones de EN candidatas. Un patrón es, por ejemplo, un nombre propio (como identificado por un etiquetador POS) seguido por un sintagma nominal en aposición (por ejemplo, "Maury Cooper, a vice president at S&P"). Los patrones se guardan en pares {spelling, context} donde "spelling" se refiere al nombre propio y "context" se refiere al sintagma nominal en el contexto. Comenzando con cimientes de reglas de ortografía, los candidatos son examinados. Los candidatos que satisfacen una regla de "ortografía" son clasificados de acuerdo a esta, y sus "contextos" son acumulados. Los contextos más frecuentes encontrados son convertidos en un conjunto de reglas

contextuales. Siguiendo estos pasos, las reglas contextuales pueden usarse para encontrar más reglas de ortografía, y así sucesivamente. Estos autores demostraron que la idea que aprender muchos tipos de EN simultáneamente permite encontrar evidencia negativa (un tipo contra todos) y reduce la sobre-generación.

Los autores muestran que el uso de datos no etiquetados puede reducir drásticamente la necesidad de supervisión a solo 7 reglas "semilla". La única supervisión que usa este enfoque es en la forma de 7 reglas "semilla" (a saber, que New York, California y U.S. son lugares; que cualquier nombre que contiene Mr. es una persona; que cualquier nombre que contiene "*Incorporated*" es una organización; y que IBM y Microsoft son organizaciones). Los resultados obtenidos están cercanos al 91% de precisión.

El enfoque se apoya en la redundancia natural de los datos: por muchas instancias de ENs la ortografía y el contexto en el que aparece son suficientes para determinar su tipo.

Como trabajo futuro los autores se plantearon extender el enfoque para construir un sistema de EEN completo, lo cual fue realizado más adelante por Kozareva et al. usando las técnicas de *self-training* y *co-training* [73]. Este enfoque difiere del anterior por el algoritmo de *co-training* que usan, por los métodos de clasificación con los que trabajan y por los conjuntos de características que usan para los módulos de REN y CEN.

Cucerzan y Yarowsky también usaron un algoritmo de aprendizaje minimalmente supervisado (*bootstrapping*) que utiliza listas cortas de nombres como datos semilla y evidencia morfológica y contextual para crear estructuras de tipo *trie* para modelar la estructura de diferentes tipos de ENs [71]. Utilizan una técnica similar a la usada por Collins y Singer y la aplican a muchos lenguajes [72].

El sistema de Riloff y Jones se caracteriza como semi-supervisado porque aprende de un pequeño número de ejemplos etiquetados y de un considerable volumen de datos no etiquetados. El objetivo es inducir un conjunto de patrones de extracción de información, los cuales pueden usarse para identificar y clasificar ENs en un texto. El sistema comienza generando exhaustivamente todos los patrones de extracción candidatos, usando un sistema llamado AutoSlog. Adicionalmente, un pequeño número de ejemplos de ENs se proveen al sistema. El patrón más útil para reconocer los ejemplos semilla es seleccionado y usado para expandir el conjunto de ENs clasificadas. Este proceso es repetido por un número de iteraciones predefinido. El resultado final es un diccionario de ENs y los patrones de extracción que les corresponden [74].

Esta es una versión más débil de la idea de Collins y Singer. En vez de trabajar con candidatos de EN predefinidos (encontrados usando una construcción sintáctica fija), ellos comienzan con un puñado de ejemplos de entidades semilla de un tipo dado (por ejemplo, Bolivia, Guatemala y Honduras son entidades de tipo "país") y acumulan todos los patrones encontrados alrededor de estas semillas en un extenso corpus. Los contextos (por ejemplo, oficinas en X, instalaciones en X, etc.) son rankeados y usados para encontrar nuevos ejemplos. Los autores de este paper apuntan que el rendimiento de este algoritmo puede deteriorarse rápidamente cuando penetra ruido en la lista de entidades o en la lista de patrones.

2.2.3 Métodos no supervisados

Los sistemas que usan este tipo de algoritmos no necesitan ejemplos de entrenamiento. El aprendizaje no supervisado no es un enfoque muy popular para la EEN debido a que los sistemas que lo usan no han alcanzado el nivel de desempeño de los sistemas supervisados. Por

este motivo, los sistemas que usan este enfoque usualmente no son completamente no supervisados, sino que tienden a ser sistemas híbridos (los cuales se tratarán en el próximo epígrafe) que combinan módulos de aprendizaje supervisado y módulos no supervisados que generan un conjunto de reglas. En este enfoque el objetivo del programa es construir representaciones de los datos, por ejemplo, clusterizando documentos similares o entidades. [5, 64].

Este tipo de enfoque puede ser fácilmente portado a diferentes dominios y lenguajes.

El enfoque típico en aprendizaje no supervisado es clustering. Por ejemplo, se puede tratar de obtener ENs de grupos clusterizados basados en la similitud del contexto. Básicamente, las técnicas recaen en recursos léxicos (por ejemplo, WordNet), en patrones léxicos y en estadísticas calculadas en un corpus grande no anotado [75].

Alfonseca y Manandhar estudian el problema de etiquetar una palabra de entrada con un tipo de EN apropiado. Los tipos de EN son tomados de WordNet (por ejemplo, location>country, animate>person, animate>animal, etc.). El enfoque es asignar una firma de tópico a cada synset de WordNet simplemente listando las palabras que co-ocurren frecuentemente en un corpus. Entonces, dada una palabra de entrada en un documento dado, el contexto de la palabra (las palabras que aparecen en una ventana de tamaño fijo alrededor de la palabra) se compara con las firmas de tipo y clasificada con la más similar [76].

En el trabajo presentado por Evans, el método para la identificación de hipónimos/hiperónimos descrito en el trabajo de Hearst en 1992 se aplica para identificar hiperónimos potenciales de secuencias de palabras capitalizadas que aparecen en un documento. Por ejemplo, cuando X es una secuencia capitalizada, la consulta "such as X" se busca en la web y, en los documentos recuperados, el sustantivo que precede inmediatamente a la consulta puede escogerse como el hiperónimo de X [77].

Shinyama et al. observaron que las ENs a menudo aparecen en muchos artículos de noticias sincronizadamente, mientras que los sustantivos comunes no. Encontraron una correlación fuerte entre ser una EN, y la aparición de forma intermitente y simultáneamente en múltiples fuentes de noticias. Esta técnica permite identificar ENs poco comunes en una forma no supervisada, y puede ser útil cuando se combina con otros métodos de EEN [78].

A su vez, Nadeau et al. usan una estrategia no supervisada usando gazetteers generados automáticamente y resolviendo después la ambigüedad. Se reconocen ENs de tipo ORG, LOC y PER. El sistema utiliza un algoritmo de generación de diccionarios y se generan reglas mediante recuperación de información. Se propone un paso de desambiguación uniendo los tipos de las ENs que tengan un alias y la asignación de la clase de la EN cuyo contexto no sea ambiguo. Este sistema no supervisado no llega a ser competitivo con sistemas supervisados, al alcanzar un 70% de valor de la medida F [75].

En este sistema hay 2 módulos: el primero es usado para crear gazetteers de entidades, tales como listas de ciudades y el segundo módulo usa heurísticas simples para identificar y clasificar entidades en el contexto de un documento dado (es decir, desambiguación de entidades).

Cucchiarelli y Velardi presenta una técnica estadística que usa un corpus de aprendizaje para adquirir pistas contextuales de clasificación, y entonces usa el resultado de esta fase para clasificar nombres propios no reconocidos. Los ejemplos de entrenamiento son obtenidos usando cualquier reconocedor de EN disponible (en los experimentos usaron un reconocedor basado en reglas y un reconocedor basado en aprendizaje automático) [79].

El enfoque descrito es complementario a los métodos actuales para el reconocimiento de EN: el objetivo de los autores es mejorar, sin esfuerzos manuales adicionales, la robustez de

cualquier sistema de EN mediante el uso de conocimiento contextual más refinado, mejor explotado en una etapa relativamente tardía del análisis. El método es particularmente útil cuando un sistema de EN debe ser rápidamente adaptado a otro lenguaje o dominio.

2.3 Enfoque híbrido

La estrategia detrás de los sistemas híbridos consiste en intentar superar las debilidades de los sistemas creados manualmente por expertos y las de los basados en aprendizaje automático de forma tal que se aprovechen las fortalezas de cada uno logrando un balance entre estos 2 enfoques. Una gran parte de los sistemas analizados son hasta cierto punto híbridos pues aunque se basen en modelos de aprendizaje automático, hacen uso de recursos lingüísticos como gazetteers o listas de palabras. A pesar de que este tipo de enfoque puede obtener mejores resultados, la debilidad de los sistemas de EEN hechos a mano permanece cuando hay necesidad de cambiar el dominio de los datos en algunos casos.

El sistema presentado por Mikheev et al., llamado LTG, hacía uso de evidencia interna y externa. Los autores se basaban en la idea que ciertas cadenas tienen una estructura que sugiere que son ENs, pero no sugieren de qué tipo y el uso de listas de nombres (de personas, lugares, etc.) puede no ser de ayuda siempre, ya que existen casos en que un nombre puede ocurrir en múltiples de ellas. Sin embargo, en algún lugar del texto es probable que haya algún tipo de información contextual que aclare qué tipo de EN es. La idea general de los autores es retrasar la clasificación final de una EN hasta que esa pieza de información contextual sea encontrada [80].

Entre las herramientas usadas en este sistema utilizan un tokenizador y una herramienta de etiquetamiento POS, entre otras.

En este sistema, las categorías TIMEX y NUMEX son manejadas de forma diferente a ENAMEX. La razón es que las expresiones numéricas y temporales en los periódicos en inglés pueden capturarse mediante reglas gramaticales por lo que desarrollaron gramáticas para estas expresiones, y también compilaron listas de entidades temporales y monedas.

Las expresiones ENAMEX, como se ha visto, son más complejas y más dependientes del contexto. El sistema hace uso de listas de organizaciones y lugares, pero son alteradas dinámicamente: si en el texto se encuentra suficiente contexto para decidir que una palabra es usada como el nombre de una organización, se añade a la lista de organizaciones para el posterior procesamiento de ese texto, por ejemplo. Cuando se comienza a procesar otro texto, no se hace ninguna suposición sobre si la palabra es una organización o lugar, hasta que se encuentre contexto suficiente para tomar esa decisión.

El sistema LTG realiza el procesamiento en 5 etapas.

En la primera etapa se utiliza un pequeño conjunto de expresiones regulares muy generales y confiables. Estas son reglas muy orientadas al contexto. Solo si una palabra ocurre en un contexto no ambiguo y no hay evidencia contradictoria (como la presencia en más de una lista), se marcará como una EN. En esta etapa, cada asignación se considera probable y no definitiva.

En la segunda etapa se define un conjunto de reglas más débiles, y al final se hace uso de un modelo de entropía máxima que tiene en cuenta diferente información contextual para tomar una decisión.

La tercera etapa está basada principalmente en gazetteers de nombres de personas, organizaciones y lugares. En esta etapa se aplican las reglas de la primera etapa, pero se relajan las restricciones de contexto. También se intenta resolver el problema de las conjunciones en los nombres de organizaciones. El sistema chequea si las posibles partes de la conjunción fueron

usadas en el texto por sí mismas como nombres de diferentes organizaciones; si no, el sistema no tiene razón para asumir que se habla de más de 1 compañía.

La cuarta etapa se basa en modelos de ME, el cual manipula reglas basadas en gazetteers.

Por último, la quinta etapa está destinada a identificar ENs en títulos, que por lo general tienen diferentes reglas de capitalización.

Este sistema logró una medida F de 93.39 por lo que fue la mayor puntuación de los sistemas de EEN participantes.

Este sistema no realiza decisiones basándose fuertemente en listas o gazetteers sino que trata la información de tales listas como "posibles" y se concentra en encontrar contextos en los cuales tales expresiones sean definitivas. El sistema aplica reglas simbólicas y técnicas de macheo parciales estadísticas de forma intercalada.

Un aspecto negativo es que en la evaluación de MUC el sistema no era rápido, solo analizaba cerca de 8 palabras por segundo, aunque en la actualidad según afirman los autores, se ha mejorado considerablemente este aspecto.

Otro sistema de este tipo con muy buen rendimiento es MENE, el cual utiliza un clasificador de entropía máxima (ME) para combinar las salidas de varios sistemas hechos a mano y obtiene resultados superiores a los obtenidos por cada uno de dichos sistemas independientemente [3].

El etiquetamiento de EN se considera como un problema de etiquetamiento de secuencias, donde múltiples características internas y externas, locales y globales son desarrolladas y combinadas. Estas características incluyen, entre otras, características léxicas, características de diccionario (se chequea si la palabra pertenece a alguna lista de nombres propios precompilada), características externas al sistema, es decir, salidas de ENs de otros sistemas y características de resolución de referencias de largo rango, es decir, nombres parciales referentes a la misma entidad.

En este sistema de Borthwick el procesamiento se hace en etapas. En la fase inicial, el texto pasa a través de reglas de expresiones regulares hechas manualmente. Estas son reglas que fueron estimadas como que tenían una muy alta probabilidad de estar correctas.

En la próxima etapa, un conjunto de reglas más débiles son pasadas a un modelo de entropía máxima. Estas reglas toman en cuenta reglas como si una palabra fue identificada o no como una EN en cualquier lugar en el texto por una de las reglas anteriores, información de caso, la posición de la palabra en la oración, etc. Los autores se basan en la idea de que cada uno de los de arriba puede ser una característica diferente que pretende dársele un peso por un procedimiento de entrenamiento de ME similar al de MENE.

Hay otro estado en el cual se usan reglas hechas a mano. Estas reglas tienen criterios más relajados que las reglas anteriormente usadas, por tanto tienen menor precisión. Estas reglas hacen uso de listas de lugares conocidos, organizaciones y nombres de personas. Le sigue otro estado de ME similar al descrito anteriormente y finalmente otro modelo ME que maneja los nombres encontrados en los encabezados de los documentos.

Este sistema puede ser usado como una herramienta de post-procesamiento para mejorar la salida de reconocedores de EN hecho a mano pues en las pruebas realizadas se combinaron 3 reconocedores externos bajo MENE, y se obtuvieron resultados que, en algunos casos, están cerca del rendimiento humano.

Otro sistema híbrido para la EEN fue el presentado por Srihari et al., los cuales combinan ME, HMM y reglas gramaticales hechas a mano. Cada uno de estos métodos tiene fortalezas y debilidades innatas pero la combinación resultó en un etiquetador de alta precisión. Este sistema además incluye gazetteers externos [81].

Los autores se basaron en el hecho de que la tarea demuestra características que pueden ser explotadas por las 3 técnicas. Por ejemplo, las expresiones de tiempo y monetarias son bastante predecibles y por tanto procesadas más eficientemente con reglas gramáticas hechas a mano. Sin embargo, las entidades de tipo ENAMEX son altamente variables y por tanto se prestan para algoritmos de entrenamiento estadísticos tales como HMM.

El primer módulo del sistema es un etiquetador basado en reglas que contiene reglas de macheo de patrones para expresiones de tipo NUMEX y TIMEX. Estas etiquetas incluyen las etiquetas estándares de MUC, así como otras muchas sub-categorías definidas por la organización.

El resto de los módulos se enfocan en el resto de las categorías (ENAMEX). El segundo módulo asigna etiquetas tentativas de personas y lugares basado en gazetteer externos de personas y lugares. En vez de confiar en la simple búsqueda en el gazetteer, lo cual es muy propenso a errores, este módulo emplea ME para construir un modelo estadístico que incorpora gazetteers con información contextual común. El módulo central del sistema es un HMM basado en bigram muy similar al de Nymble [9]. Fueron diseñadas unas reglas con el objetivo de corregir errores en la segmentación de ENs, las cuales fueron incorporadas en un HMM. Estas reglas sirven como restricciones en el modelo HMM y les permiten utilizar información más allá de los bigram y eliminar los errores obvios debido a la limitación del cuerpo de entrenamiento. El HMM genera las etiquetas estándar de MUC, persona, lugar y organización. Basado en ME, el último módulo deriva sub-categorías tales como ciudad, aeropuerto, gobierno, etc. de las etiquetas base.

Las reglas implementadas incluyen una gramática para expresiones temporales, una gramática para expresiones numéricas y una gramática para otras ENs no-MUC (por ejemplo: información de contacto como dirección, email).

El sistema fue evaluado con los datos de MUC-7 con los cuales se obtuvo una medida F de 93.39.

Cruz presentó un enfoque híbrido a la EEN para el idioma español que combina clasificadores secuenciales markovianos con un conjunto de heurísticas en forma de expresiones regulares [82].

El reconocedor recibe como entrada una secuencia de palabras etiquetadas con su información morfosintáctica, es decir, su parte de la oración (etiquetamiento POS) y sus características morfológicas, y asigna una etiqueta a cada una.

El sistema está compuesto por dos unidades: un clasificador secuencial y un conjunto de heurísticas. El clasificador secuencial se entrena a partir de un corpus y produce una primera secuencia de etiquetas. Cada heurística es definida mediante una expresión regular y se encarga de corregir las posibles omisiones de ENs.

Como clasificadores secuenciales utilizan dos tipos de modelos markovianos, los HMM, y los PMM (del inglés *Projection Markov Models*). En el modelo las palabras son vistas como vectores de rasgos binarios definidos. De esta forma, se obtiene un conjunto de observaciones finito, el cual se restringe a los posibles vectores binarios que se pueden formar. Con esto se pretende que el modelo no memorice las palabras sino que más bien se centre en las "situaciones".

Adicionalmente, se estima la probabilidad de clasificación de un vector de rasgos $< f_1, \ldots, f_m >$ con una etiqueta c_i, o sea, $P(c = ci \mid f1 = v1, \ldots, fm = vm)$ y se utilizan las probabilidades obtenidas de esta forma en un combinador basado en modelos de Markov por proyecciones.

Se utilizan rasgos que se refieren a la morfología de las palabras (por ejemplo, si la palabra comienza o no con mayúscula), a su información morfosintáctica (parte de la oración, etc.), a su pertenencia a algún diccionario, así como rasgos léxicos.

Los diccionarios que se utilizan contienen títulos de personas (por ejemplo: Sr. o Ing.), nombres de personas, apellidos, cargos, tipos de organizaciones (por ejemplo: Federación o Asociación), nombres de ciudades, países, nombres de monedas, los meses del año, etc. Por su parte, mediante los rasgos léxicos se chequea la palabra que se analiza, la palabra que le antecede o la que se encuentra dos posiciones por delante de ella. Mediante éstos se chequea la ocurrencia de palabras sintácticamente importantes, tales como conjunciones o preposiciones que indican pertenencia, origen, destino, etc.

Las heurísticas utilizadas tienen la forma de expresiones regulares sobre el alfabeto formado por los rasgos binarios definidos sobre las palabras. Estas heurísticas pueden lograr un alto grado de generalidad. Debe notarse que, a pesar de la generalidad de las heurísticas, éstas no deben utilizarse como único recurso en la solución del problema de la EEN ya que éstas no pueden resolver los casos de ambigüedad como la existente entre los nombres de personas y los nombres de lugares y entre éstos y los nombres de organizaciones.

Con el objetivo de mostrar el efecto de la combinación de los clasificadores secuenciales y las heurísticas, así como evaluar la importancia de diferentes tipos de rasgos, se construyó un conjunto de reconocedores usando distintos tipos de clasificadores secuenciales y combinaciones de rasgos.

En los experimentos realizados se obtuvo como resultado que la utilización de las expresiones regulares provoca un descenso en el valor de la precisión. Esto se debe a la clasificación incorrecta de algunas ENs que habían sido omitidos. Sin embargo, el valor del recall en todos los casos aumenta debido a la recuperación de una mayor cantidad de nombres mediante las expresiones regulares. Dado que el aumento del recall es mayor que la disminución de la precisión, el valor de la medida F1 aumenta en todos los casos.

Los resultados experimentales demuestran que las medidas de calidad de un sistema para la tarea de EEN en español basado en clasificadores secuenciales probabilísticos aumentan con la integración de conocimiento lingüístico mediante expresiones regulares. Además, estos resultados permiten corroborar que la elección de los rasgos que se utilizan para representar las palabras influye notablemente en la calidad del etiquetado.

2.4 Resumen de resultados obtenidos

En la Tabla 8 se presentan los resultados obtenidos por diversos sistemas en varias evaluaciones y para varios idiomas.

Tabla 8: Resultados de los sistemas

Sistema	Idioma	Conferencia	F
E. Sang, 2002 [26]	Español/Holandés	CoNLL-2002	75.78/70.67
Hendrickx y Bosch, 2003 [67]	Inglés/Alemán	CoNLL-2003	78.20/63.02
RoboTag [48]	Japonés/Inglés	MET-1/MUC-6	83,6/88,1
New York University	Inglés/Japonés	MUC-7	88.80/79.51
FASTUS [49]	Inglés	MUC-6	94

LOLITA [53]	Inglés	MUC-6	67.62
Nymble [9]	Inglés/Español	MUC-6/MET-1	93/90
Srihari y Li, 2001 [81]	Inglés	MUC-7	89
Paliouras et al, 2000 [57]	Inglés	MUC-6	85
Facile [51]	Inglés	MUC-7	81,91
MENE [5]			92,2
Meulder et al, 2003 [83]	Inglés/Alemán	CoNLL-2003	76,97/57,27
McNamee et al, 2002 [66]	Español/Holandés	CoNLL-2002	60.97/59,52
Carreras et al, 2002a [21]	Español/Holandés	CoNLL-2002	81.39/77.05
Florian, 2002 [22]	Español/Holandés	CoNLL-2002	79.05/74.99
Cucerzan y Yarowsky, 2002 [23]	Español/Holandés	CoNLL-2002	77.15/72.31
LTG [80]	Inglés	MUC-7	93.39
IsoQuest [52]	Inglés	MUC-7	91.6
IsoQuest system 1	Inglés	MUC-7	91.60
IsoQuest system 2	Inglés	MUC-7	82.61
Zhou y Su, 2002 [47]	Inglés	MUC-6/MUC-7	96.6/94.1
Wu et al, 2002 [24]	Español/Holandés	CoNLL-2002	76.61/75.36
Burger et al, 2002 [25]	Español/Holandés	CoNLL-2002	75.78/72.57
Patrick et al, 2002 [27]	Español/Holandés	CoNLL-2002	73.92/71.36
Jansche, 2002 [28]	Español/Holandés	CoNLL-2002	73.89/69.68
Malouf, 2002 [29]	Español/Holandés	CoNLL-2002	73.66/68.08
Tsukamoto et al, 2002 [30]	Español/Holandés	CoNLL-2002	71.49/60.93
Black et al, 2002 [84]	Español/Holandés	CoNLL-2002	63.73/49.75
Florian et al, 2003 [31]	Inglés/Alemán	CoNLL-2003	88.76±0.7/72.41±1.3
Chieu et al, 2003 [32]	Inglés/Alemán	CoNLL-2003	88.31±0.7/65.67±1.4
Klein et al, 2003 [33]	Inglés/Alemán	CoNLL-2003	86.07±0.8/71.90±1.2
Zhang et al, 2003 [34]	Inglés/Alemán	CoNLL-2003	85.50±0.9/71.27±1.5
Carreras et al 2003a [35]	Inglés/Alemán	CoNLL-2003	85.00±0.8/69.15±1.3
Curran et al, 2003 [36]	Inglés/Alemán	CoNLL-2003	84.89±0.9/68.41±1.4
Mayfield et al, 2003 [37]	Inglés/Alemán	CoNLL-2003	84.67±1.0/69.96±1.4
Carreras et al 2003b [38]	Inglés/Alemán	CoNLL-2003	84.30±0.9/66.48±1.5
McCallum et al, 03 [39]	Inglés/Alemán	CoNLL-2003	84.04±0.9/68.11±1.4
Bender et al, 2003 [40]	Inglés/Alemán	CoNLL-2003	83.92±1.0/68.88±1.3
Munro et al, 2003 [41]	Inglés/Alemán	CoNLL- 2003	82.50±1.0/67.75±1.4
Wu et al, 2003 [42]	Inglés/Alemán	CoNLL-2003	81.70±0.9/66.

			34±1.3
Whitelaw et al, 2003 [43]	Inglés/Alemán	CoNLL-2003	79.78±1.0/54.43±1.4
Hammerton, 2003 [85]	Inglés/Alemán	CoNLL-2003	60.15±1.3/47.74±1.5
Kent Ridge Digital Labs Official Summary Scores	Chino	MUC-7	86.38
National Taiwan University	Chino	MUC-7	79.61
NTT	Japonés	MUC-7	83.72
OKI	Japonés	MUC-7	90.54
Kent Ridge Digital Labs (NUS)	Inglés	MUC-7	77.74
The MITRE Corporation	Inglés	MUC-7	85.31
National Taiwan University	Inglés	MUC-7	69.67
OKI	Inglés	MUC-7	84.05
University of Durham	Inglés	MUC-7	76.43
University of Manitoba system 1	Inglés	MUC-7	86.37
University of Manitoba system 2	Inglés	MUC-7	83.70
University of Sheffield	Inglés	MUC-7	85.83

3 Conclusiones

La tarea de EEN es abordada por muchos enfoques, tanto por métodos hechos a mano, por métodos de *machine learning* e incluso por enfoques híbridos. Para utilizar algoritmos de aprendizaje supervisado es necesario contar con un corpus etiquetado con los nombres de entidades que sirvan como ejemplos para el entrenamiento. La tarea de etiquetar este corpus también requiere un esfuerzo considerable, sin embargo, el grado de especialización del personal dedicado a ella no necesita ser tan alto como el del personal que define los conjuntos de reglas. En la práctica, los sistemas basados en aprendizaje supervisado han obtenido mejores resultados que aquellos basados en aprendizaje no supervisado.

Según Feldman podemos ver el problema de EEN como un problema de clasificación, donde clasificamos cada palabra como perteneciente a una de las clases de EN o a la clase 'no-name'. Una de las técnicas más populares para tratar con la clasificación de secuencias es HMM [58].

Un aspecto muy importante mencionado por varios proyectos es que de acuerdo a McDonald hay 2 tipos complementarios de evidencia: la interna la cual se toma dentro de una EN, y externa, la cual se provee por el contexto en el cual aparece el nombre. Para algunas palabras es cierto que siempre son parte de, están seguidas o precedidas por una EN particular. Para otras, solo a veces ocurre esto. Ejemplos del primer caso son "Mr.", "Prof.", seguidos por un nombre propio. Ejemplo de lo otro son los nombres de compañías seguidos por "produce" [86].

De forma general para tener en cuenta la evidencia externa se tienen en cuenta varias palabras alrededor del token a analizar. Todos los sistemas que hacen uso de esto no toman el mismo tamaño de ventana.

Se puede evidenciar además que una característica importante para la tarea de EEN es la información relativa a la capitalización de las palabras como fue observado por Bikel et al., Borthwick y Miller et al. [9], [3], [59].

Las características usadas por los sistemas es un factor muy importante en el rendimiento de los mismos pues según Curran y Clark el relativamente bajo rendimiento de los sistemas usados en CoNLL-2002 fue mayormente debido a los conjuntos de características usados más que al método de aprendizaje, lo cual es evidenciado por el buen rendimiento del sistema de Zhou y Su [47], quienes usan gran variedad de características [36].

Sánchez hizo un sistema que tenía en cuenta todas las apariciones de las ENs en el texto para poder hacer uso de un contexto global en vez de un contexto local. El sistema no tuvo muy buenos resultados, pero la idea de aprovechar la información de cada aparición de la EN en el texto es acertada [4].

Algunos autores han usado algoritmos de alias para saber cuándo una EN está haciendo referencia a otra.

A pesar de que la tendencia creciente en los sistemas de EEN es utilizar formalismos del aprendizaje automático es conveniente la integración de conocimiento lingüístico a estos sistemas debido a la simplicidad y generalidad que puede tener el mismo [82].

El uso de gazetteers parece inevitable pues los experimentos realizados por diversos autores evidenciaron el efecto positivo de usar información de este tipo.

Referencias bibliográficas

1. Rau, L.F., *Extracting Company Names from Text.* 1991.
2. *Definitions of terms used in Information Extraction*. 2005; Available from: http://www-nlpir.nist.gov/related_projects/muc/info/definitions.html.
3. Borthwick, A. *A Maximum Entropy approach to Named Entity Recognition*. 1999.
4. Sánchez, C.R., *Clasificación de Entidades Nombradas utilizando Información Global*. 2008.
5. Mansouri, A., L.S. Affendey, and A. Mamat, *Named Entity Recognition Approaches.* 2008.
6. Cruz, Y.R., *Métodos para el Reconocimiento de Nombres de Entidades Anidados y No Anidados en Español*. 2008.
7. Carreras, X., L. Marquez, and L. Padro. *Wide-Coverage Spanish Named Entity Extraction*. 2002.
8. Rössler, M. *Using Markov Models for Named Entity recognition in German newspapers*. 2002.
9. Bikel, D.M., et al., *Nymble: a High-Performance Learning Name-finder.* 1997.
10. *Evaluación de la recuperación de documentos.* 12/06/2009]; Available from: http://evalua-recuperacion.tripod.com/muc.html.
11. *Evaluación de la recuperación de documentos*. 12/06/09]; Available from: http://evalua-recuperacion.tripod.com/muc.html.
12. *MUC Evaluations*. Available from: http://www-nlpir.nist.gov/related_projects/muc/index.html.
13. *MUC-6*. [cited 2009 10/06/09]; Available from: http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html.
14. *The Message Understanding Conference Scoring Software User's Manual*. 1995; Available from: http://www-nlpir.nist.gov/related_projects/muc/muc_sw/muc_sw_manual.html.
15. Sundheim, B.M., *Overview of results of the MUC-6 evaluation.* 1995.
16. *Named Entity Task Definition*. 1998; Available from: http://www.cs.nyu.edu/cs/faculty/grishman/NEtask20.book_2.html#HEADING1.
17. *Named Entity Scores - Chinese*. 1998; Available from: http://www-nlpir.nist.gov/related_projects/muc/proceedings/ne_chinese_score_report.html.
18. *Named Entity Scores - English*. 1998; Available from: http://www-nlpir.nist.gov/related_projects/muc/proceedings/ne_english_score_report.html.
19. *Statistical Significance Results*. 1998; Available from: http://www-nlpir.nist.gov/related_projects/muc/proceedings/stat_sig_index.html.

20. *Language-Independent Named Entity Recognition (I).* 2005 [cited 2009 23/06/2009]; Available from: http://www.cnts.ua.ac.be/conll2002/ner.
21. Carreras, X., L. Marquez, and L. Padro. *Named entity extraction using adaboost.* 2002.
22. Florian, R., *Named Entity Recognition as a House of Cards: Classifier Stacking.* 2002.
23. Cucerzan, S. and D. Yarowsky, *Language Independent NER using a Unified Model of Internal and Contextual Evidence.* 2002.
24. Wu, D., et al. *Boosting for named entity recognition.* 2002.
25. Burger, J.D., J.C. Henderson, and W.T. Morgan, *Statistical Named Entity Recognizer Adaptation.* 2002.
26. Sang, E.F.T.K., *Memory-Based Named Entity Recognition.* 2002.
27. Patrick, J., C. Whitelaw, and R. Munro, *SLINERC: The Sydney Language-Independent Named Entity Recogniser and Classifier.* 2002.
28. Jansche, M., *Named Entity Extraction with Conditional Markov Models and Classifiers.* 2002.
29. Malouf, R., *Markov models for language-independent named entity recognition.* 2002.
30. Tsukamoto, K., Y. Mitsuishi, and M. Sassano, *Learning with Multiple Stacking for Named Entity Recognition.* 2002.
31. Florian, R., et al., *Named Entity Recognition through Classifier Combination.* 2003.
32. Chieu, H.L. and H.T. Ng, *Named Entity Recognition with a Maximum Entropy Approach.* 2003.
33. Klein, D., et al., *Named Entity Recognition with Character-Level Models.* 2003.
34. Zhang, T. and D. Johnson, *A Robust Risk Minimization based Named Entity Recognition System.* 2003.
35. Carreras, X., L. Márquez, and L. Padro. *A Simple Named Entity Extractor using AdaBoost.* 2003.
36. Curran, J.R. and S. Clark, *Language Independent NER using a Maximum Entropy Tagger.* 2003.
37. Mayfield, J., P. McNamee, and C. Piatko, *Named Entity Recognition using Hundreds of Thousands of Features.* 2003.
38. Carreras, X., L. Màrquez, and L. Padró, *Learning a Perceptron-Based Named Entity Chunker via Online Recognition Feedback.* 2003.
39. McCallum, A. and W. Li, *Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons.* 2003.
40. Bender, O., F.J. Och, and H. Ney, *Maximum Entropy Models for Named Entity Recognition.* 2003.
41. Munro, R., D. Ler, and J. Patrick, *Meta-Learning Orthographic and Contextual Models for Language Independent Named Entity Recognition.* 2003.
42. Wu, D., G. Ngai, and M. Carpuat, *A Stacked, Voted, Stacked Model for Named Entity Recognition.* 2003.
43. Whitelaw, C. and J. Patric, *Named Entity Recognition Using a Character-based Probabilistic Approach.* 2003.
44. Grishman, R. and B. Sundheim, *Message Understanding Conference-6: A Brief History.* 2006.
45. Sekine, S., K. Sudo, and C. Nobata, *Extended Named Entity Hierarchy.* 2002.
46. Sekine, S. and C. Nobata, *Definition, dictionaries and tagger for Extended Named Entity Hierarchy.* 2004.
47. Zhou, G. and J. Su. *Named Entity Recognition using an HMM-based Chunk Tagger.* 2002.
48. Bennett, S.W., C. Aone, and C. Lovell, *Learning to Tag Multilingual Texts Through Observation.* 1997.
49. Appelt, D.E., et al., *Sri international FASTUS system MUC-6 test results and analysis.* 1995.
50. Grishman, R., *The NYU system for MUC-6 or where's the syntax?* 1995.
51. Black, W.J., F. Rinaldi, and D. Mowatt, *Facile: description of the ne system used for MUC-7.* 1998.
52. Krupka, G.R. and K. Hausman, *IsoQuest, Inc.: Description of the NetOwl™ Extractor System as Used for MUC-7.* 1998.

53. Morgan, R., et al., *University of Durham: Description of the LOLITA system as used in MUC-6.* 1995.
54. Bikel, D.M., R. Schwartz, and R.M. Weischedel, *An Algorithm that Learns What's in a Name.* 1999.
55. Sekine, S., *Description of the Japanese NE System Used For MET-2.* 1998.
56. Sekine, S., R. Grishman, and H. Shinnou, *A Decision Tree Method for Finding and Classifying Names in Japanese Texts.* 1998.
57. Paliouras, G., et al., *Learning Decision Trees for Named-Entity Recognition and Classification.* 2000.
58. Feldman, R., *Information Extraction. Theory and Practice (ICML 2006).* 2006.
59. Miller, S., et al., *Algorithms that learn to extract information. BBN: description of the SIFT system as used for MUC-7.* 1998.
60. Borthwick, A., et al., *NYU: Description of the MENE Named Entity System as Used in MUC-7.* 1998.
61. Berger, A.L., S.A.D. Pietra, and V.J.D. Pietra, *A maximum entropy approach to natural language processing.* 1996.
62. Ratnaparkhi, A., *A simple introduction to maximum entropy models for natural language processing.* 1997.
63. Budi, I. and S. Bressan, *Association Rules Mining for Name Entity Recognition.* Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003.
64. Bogers, T., *Dutch Named Entity Recognition: Optimizing Features, Algorithms, and Output*. 2004.
65. Milidiú, R.L., J.C. Duarte, and R. Cavalcante, *Machine Learning Algorithms for Portuguese Named Entity Recognition.* 2007.
66. McNamee, P. and J. Mayfield, *Entity extraction without language-specific resources.* 2002.
67. Hendrickx, I. and A.v.d. Bosch, *Memory-based one-step named-entity recognition: Effects of seed list features, classifier stacking, and unannotated data.* 2003.
68. Carreras, X., L. Marquez, and L. Padro. *A Simple Named Entity Extractor using AdaBoost*. 2003.
69. Rifkin, R. and A. Klautau, *In Defense of One-Vs-All Classification.* 2004.
70. Sekine, S., *Named Entity: History and Future* 2004.
71. Cucerzan, S. and D. Yarowsky, *Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence.* 1999.
72. Collins, M. and Y. Singer, *Unsupervised Models for Named Entity Classification.* 1999.
73. Kozareva, Z., B. Bonev, and A. Montoyo, *Self-training and Co-training Applied to Spanish Named Entity Recognition.* 2005.
74. Riloff, E. and R. Jones, *Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping.* 1999.
75. Nadeau, D., P. Turney, and S. Matwin, *Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity.* 2006.
76. Alfonseca, E. and S. Manandhar, *An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery.* 2002.
77. Evans, R., *A Framework for Named Entity Recognition in the Open Domain.* 2003.
78. Shinyama, Y. and S. Sekine, *Named Entity Discovery Using Comparable News Articles.* 2004.
79. Cucchiarelli, A. and P. Velardi, *Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence.* 2001.
80. Mikheev, A., C. Grover, and M. Moens, *Description of the LTG system used for MUC-7.* 1998.
81. Srihari, R. and C.N.a.W. Li, *A Hybrid Approach for Named Entity and Sub-Type Tagging.* 2001.
82. Cruz, Y.R., et al., *Un enfoque híbrido al Reconocimiento de Nombres de Entidades para el español.* 2007.
83. Meulder, F.D. and W. Daelemans, *Memory-Based Named Entity Recognition using Unannotated Data.* 2003.

84. Black, W.J. and A. Vasilakopoulos, *Language-Independent Named Entity Classification by Modified Transformation-Based Learning and by Decision Tree Induction.* 2002.
85. Hammerton, J., *Named Entity Recognition with Long Short-Term Memory.* 2003.
86. McDonald, D.D., *Internal and External Evidence in the Identification and Semantic Categorization of Proper Names.* 1996.

RT_011, marzo 2010
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: DCG Matilde Galindo Sánchez
RNPS No. 2143
ISSN 2072-6260
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. No. 21812 e/218 y 222, Rpto. Siboney, Playa;
Ciudad de La Habana. Cuba. C.P. 12200
Impreso en Cuba