Procesamiento del Lenguaje Natural, núm. 35 (2005), pp. 37-44 recibido 28-04-2005; aceptado 01-06-2005

NERUA: sistema de detección y clasificación de entidades utilizando aprendizaje automático*

Óscar Ferrández, Zornitsa Kozareva, Andrés Montoyo, Rafael Muñoz
Dept. de Lenguajes y Sistemas Informáticos (Universidad de Alicante)
Carretera San Vicente s/n 03690 Alicante España
{ofe,zkozareva,montoyo,rafael}@dlsi.ua.es

Resumen: Este artículo presenta un sistema de reconocimiento de entidades para Español combinando diferentes algoritmos de aprendizaje. Se propone una detección de entidades independiente del lenguaje y se estudia la influencia del tamaño del corpus de entrenamiento en los resultados. NERUA obtuvo 92.96 % f-score en la detección y 78.59 % en la clasificación de entidades.
Palabras clave: Reconocimiento de entidades, aprendizaje automático, independiente del lenguaje

Abstract: In this paper we present a Named Entity Recognition system developed for Spanish by combining different machine learning techniques. A language independent approach for NE detection and evaluation of the influence of the training corpus size have been made. NERUA obtained 92.96 % f-score for detection and 78.59 % for classification.
Keywords: Named entity recognition, machine learning, language independent

1. Introducción

El enorme crecimiento de información digital en la actualidad, hace necesario sistemas que la procesen, analicen y exploten. El tratamiento de esta información requiere tareas de extracción, filtrado y clasificación de información relevante. El desarrollo de estas tareas se realiza por medio de aplicaciones como Extracción de Información, Recuperación de Información, Búsquedas de Respuestas, etc. Los sistemas de reconocimiento de entidades, en inglés Named Entity Recognition (NER), se encargan de realizar una detección y clasificación de las entidades que aparecen en los textos dentro de determinadas categorías predefinidas. Dichos sistemas son considerados como un paso previo a la comprensión automática de un texto, pues aportan información relevante sobre su contenido, y normalmente se encuentran integrados en sistemas más complejos como los citados anteriormente.

En este artículo proponemos un sistema de reconocimiento de entidades para el Español. La tarea de reconocimiento de entidades la dividimos en dos, detección[1] y clasificación[2]. La clasificación realiza el etiquetado de las entidades en cuatro categorías: PER, LOC, ORG y MISC como son definidas en CoNLL-2002 (Sang, 2002). Para ambas tareas empleamos tres algoritmos de aprendizaje automático: Hidden Markov Model, Máxima Entropía y Memory-based learner, para su evaluación utilizamos los recursos proporcionados por CoNLL-2002 para el español. Además, proponemos una combinación adecuada de los clasificadores que mejore los resultados obtenidos individualmente.

Los algoritmos de aprendizaje precisan conjuntos de características para el aprendizaje y la posterior clasificación. Por tal motivo, realizamos un estudio sobre las características a emplear en las tareas de detección y clasificación y qué combinación de ellas se acopla mejor a cada clasificador. A partir de este estudio, obtenemos un conjunto de características independientes del lenguaje del texto para la detección de entidades y valoramos sus resultados con respecto a otros conjuntos dependientes del lenguaje. Resulta interesante este conjunto, ya que al ser independiente del lenguaje nos permite extender nuestro sistema sobre otras lenguas.

Por otro lado, consideramos relevante rea-

* Esta investigación ha sido parcialmente financiada bajo los proyectos CICyT número TIC2003-07158-C04-01 y PROFIT número FIT-340100-2004-14 y por la Generalitat Valenciana bajo los proyectos GV04B-276 y GV04B-268

[1] Entendemos por detección, la tarea de encontrar en el texto las entidades que se consideran relevantes

[2] Entendemos por clasificación, etiquetar las entidades detectadas dentro de diferentes categorías previamente definidas

lizar un seguimiento de la eficiencia de los clasificadores en función del número de entidades que aparecen en el texto de entrenamiento. De esta manera conoceremos la cantidad de corpus de entrenamiento necesaria para que nuestro sistema se comporte con resultados similares a los obtenidos con el corpus de entrenamiento completo. Este tamaño óptimo de corpus de entrenamiento, nos permite reducir el tiempo de procesamiento del sistema y tener constancia del tamaño necesario para futuras anotaciones de corpus u obtención de los mismos.

Este artículo está organizado de la siguiente manera. En la sección 2 describimos nuestro sistema, los clasificadores y los corpus utilizados. En la sección 3 exponemos los diferentes conjuntos de características empleados para cada tarea. La sección 4 presenta el estudio realizado para obtener un corpus de entrenamiento de tamaño reducido. La sección 5 nos muestra los resultados obtenidos y su discusión. La sección 6 desarrolla una comparativa con los sistemas mostrados en CoNLL-2002 y en la sección 7 realizamos una conclusión de nuestro trabajo y exponemos trabajos futuros.

2. Descripción del sistema

Nuestro sistema aborda la tarea de reconocimiento de entidades en varios módulos como se observa en la Figura 1. Los módulos más importantes son: el módulo NED encargado de la detección de las entidades y el módulo que realiza la clasificación de las mismas (NEC). Sin embargo, éstos módulos necesitan alimentarse de las características obtenidas del texto, dicha labor la realizan los módulos MEC para la detección y para la clasificación. Como salida el sistema produce los textos de entrada etiquetados con las entidades detectadas y clasificadas.

El módulo de detección de entidades utiliza dos modelos de aprendizaje automático, concretamente Memory-based learner y Hidden Markov Model, mientras que para la implementación del módulo de clasificación hemos empleado, además de los modelos anteriores, un modelo basado en el principio de Máxima Entropía. La razón por la cual no se ha incorporado el modelo de Máxima Entropía a la detección de las entidades radica en la gran cantidad de tiempo de procesamiento necesario por dicho modelo para tal tarea, sin obtener mejores resultados que los

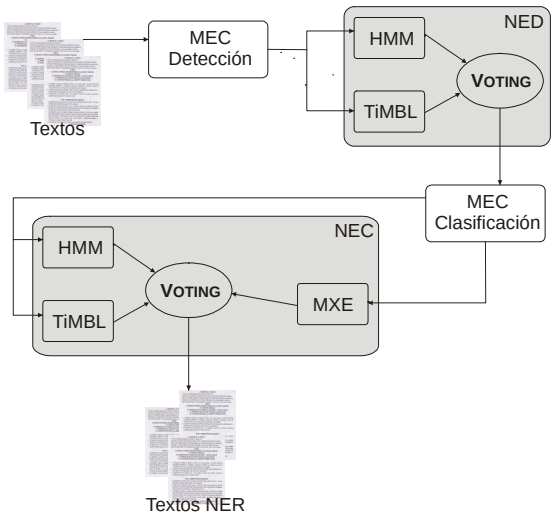

Figura 1: Descripción del sistema

otros modelos. En ambos casos, detección y clasificación de entidades, se desarrolla una estrategia de *voting* que nos permitirá aumentar los resultados mediante una cooperación adecuada de los clasificadores. En la detección, al disponer de sólo dos clasificadores, el *voting* se realiza mediante combinaciones de estos clasificadores con diferentes conjuntos de características.

2.1. Modelos de clasificación

El software empleado para los anteriores modelos han sido los siguientes: ICOPOST[3] para Hidden Markov Model, con funcionalidad para POS tagging y adaptada al reconocimiento de entidades (Schröder, 2002), el paquete software TiMBL (Daelemans et al., 2003) para Memory-based learner y el modelo usado para Máxima Entropía es el mostrado en (Suárez y Palomar, 2002).

2.2. Combinación de los clasificadores

Al disponer de varios clasificadores, resulta una práctica muy interesante y beneficiosa realizar una combinación adecuada de las salidas de cada clasificador, la cual mejore la salida de cada uno individualmente. La aproximación más simple para combinar los clasificadores es a través de un *voting*, mediante el cual se examina cada salida individual y se selecciona la clasificación que posee un peso que supere un determinado umbral, donde el peso depende de los modelos propuestos. Es posible asignar diferentes pesos para cada

[3] http://acopost.sourceforge.net/

uno de los modelos, de esta forma podemos darle mayor importancia a un modelo respecto a otro. En nuestro sistema se asignan diferentes pesos a cada modelo dependientes de la clase correcta que determina.

2.3. Corpus utilizados y su evaluación

Todas las pruebas, experimentos y resultados obtenidos, utilizan los recursos proporcionados para el español en CoNLL-2002 (Sang, 2002). El corpus de entrenamiento contiene 264725 tokens y 18794 entidades, mientras que para el realizar el test usamos el corpus Test-B que consta de 51533 tokens y 3558 entidades.

Las medidas utilizadas para la evaluación fueron Precisión (de las etiquetas asignadas por el sistema, cuantas fueron correctas), Cobertura (de las etiquetas que debería haber encontrado el sistema, cuantas encontró) y $F_{\beta=1}$(combina cobertura y precisión). Esta evaluación fue realizada con el script *Conlleval*, para así poder comparar los resultados con los sistemas de CoNLL-2202.

3. Conjunto de características

Para las tareas de detección y clasificación, los módulos de Memory-based learner y Máxima Entropía precisan un conjunto de características que les permitan clasificar las entidades existentes en el texto. A diferencia de éstos, Hidden Markov Model no utiliza estos conjuntos, sino que realizamos una especialización de las etiquetas incorporándoles información adicional como se refleja en el estudio realizado por (Rössler, 2002).

3.1. Características para la detección

Utilizamos el conocido modelo BIO para la detección, donde cada etiqueta indica si la palabra corresponde al inicio de una entidad (B), se encuentra dentro de la entidad (I) o por el contrario no es considerada como una entidad (O). Para la sentencia: *Bill Clinton afirma*, se le asociaría el siguiente conjunto de etiquetas "BIO", donde *Bill* con etiqueta B asociada representa el inicio de la entidad, *Clinton* con la etiqueta I continúa siendo parte de la entidad y el resto de palabras son señaladas con la etiqueta O como palabras no pertenecientes a ninguna entidad.

El conjunto para la detección se compone de 29 características descritas en la Figura 2. Para mejorar los resultados de los clasificadores hemos desarrollado diferentes combinaciones de este conjunto las cuales serán detalladas más adelante.

- **e**: la palabra a ser clasificada
- ***cntxt[1-6]***: palabra del contexto en posición ±1, ±2, ±3
- ***CNTXT[1-7]***: palabra en mayúsculas en la posición 0, ±1, ±2, ±3
- ***EE[1-3]***: palabra +1,+2,+3 en diccionario entre entidades
- ***pos***: posición de la palabra en la sentencia
- ***eMy***: palabra completa en mayúsculas
- ***eDic***: palabra en algún diccionario
- ***eDisp***: palabra en diccionarios de disparadores
- ***cntxtDisp***: palabra en la posición ±1, ±2, ±3 en diccionarios de disparadores
- ***eLem***: lema de la palabra
- ***eRaiz***: raíz de la palabra
- ***SubStr[1-5]***: ±2, ±3 y la mitad de caracteres de la palabra

Figura 2: Características para la detección

3.2. Características para la clasificación

Las etiquetas utilizadas para la clasificación de las entidades detectadas son PER, LOC, ORG y MISC como se definieron en CoNLL-2002. Para esta clasificación usamos las primeras siete características utilizadas para el modelo BIO, la posición de la palabra y un conjunto adicional descrito en la Figura 3. Los diccionarios utilizados para los atributos han sido recolectados aleatoriamente de diferentes páginas web.

4. Evaluación del tamaño del corpus de entrenamiento

En este apartado realizamos un estudio para determinar el tamaño óptimo del corpus de entrenamiento para los clasificadores, y así añadirle valor a nuestro sistema enriqueciéndolo con una menor necesidad de disponer de corpus de aprendizaje extensos y un tiempo de procesamiento más bajo. Troceamos el corpus de entrenamiento en función del número de entidades que alberga y para cada trozo calculamos $F_{\beta=1}$, relación entre la precisión y cobertura, que consigue cada clasificador con el trozo de corpus estudiado.

- ***eP***: es la entidad un disparador de PER
- ***eL***: es la entidad un disparador de LOC
- ***eO***: es la entidad un disparador de ORG
- ***eM***: es la entidad un disparador de MISC
- ***tP***: palabra ±1, ±2, ±3 disparador de PER
- ***tL***: palabra ±1, ±2, ±3 disparador de LOC
- ***tO***: palabra ±1, ±2, ±3 disparador de ORG
- ***gP***: parte de entidad en diccionarios PER
- ***gL***: parte de entidad en diccionarios LOC
- ***gO***: parte de entidad en diccionarios ORG
- ***wP***: entidad completa en diccionarios PER
- ***wL***: entidad completa en diccionarios LOC
- ***wO***: entidad completa en diccionarios ORG
- ***NoE***: entidad completa en ninguno de los diccionarios
- ***f***: primera palabra de la entidad
- ***s***: segunda palabra de la entidad
- ***clx***: mayúsculas, minúsculas y otros

Figura 3: Características para la clasificación

4.1. NE-detección

En la tarea de detección de entidades a medida que aumentamos el número de entidades los clasificadores mejoran sus resultados. No obstante, como nos muestra la Figura 4 cuando los modelos aprenden con aproximadamente 15000 entidades consiguen resultados muy similares a los conseguidos con todo el corpus de entrenamiento. Podemos realizar la hipótesis de que con un corpus de dicho tamaño conseguiríamos resultados similares ganando en tiempo de procesamiento y tamaño de corpus utilizado.

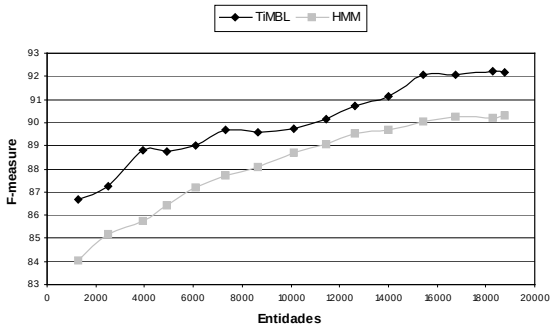

Figura 4: Evaluación del tamaño del corpus de entrenamiento para la detección

4.2. NE-clasificación

El comportamiento de los clasificadores ante la tarea de clasificación es similar. Como se muestra en la Figura 5 los modelos alcanzan resultados similares a los obtenidos con el corpus completo cuando aprenden con aproximadamente 14000 entidades.

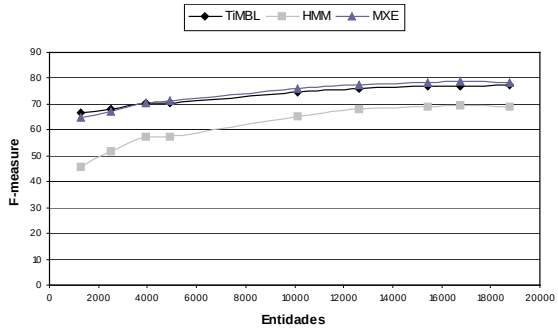

Figura 5: Evaluación del tamaño del corpus de entrenamiento para la clasificación

Para completar el estudio y comprobar que la hipótesis sobre el tamaño del corpus de entrenamiento es acertada, utilizaremos en los experimentos tanto el corpus completo, como un corpus de entrenamiento de tamaño reducido. El cual, para unificar el tamaño para las tareas de detección y clasificación, constará de 15000 entidades.

5. Experimentos y discusión

5.1. NE-detección con el corpus completo de entrenamiento

La tarea de detección de las entidades en el texto la realizamos con dos clasificadores, TiMBL y HMM. Los experimentos con TiMBL utilizan diferentes combinaciones de las características especificadas en la subsección 3.1. Los objetivos de crear diferentes conjuntos son: reducir el número de características y así disminuir el tiempo de procesamiento, encontrar aquel conjunto que mejor se acople al clasificador y tener diversas combinaciones de clasificador con conjuntos de características para poder aplicar el *voting* entre ellos.

El Cuadro 1 muestra los diferentes conjuntos de características creados para TiMBL y las características asociadas a cada uno.

Creamos tres conjuntos dependientes del lenguaje, pues utilizan diccionarios y herramientas como lematizadores y *stemmers*, ST, $STred$ y $CNTXr$. Las características usadas son similares y los resultados obtenidos con cada conjunto son mostrados en el Cuadro 2. Los mejores resultados los obtiene ST, pero analicemos un poco más el resto de conjuntos. $STred$ es un subconjunto de ST, extrayendo de ST las características que aportan mayor información, y aunque sus resultados totales

Conjunto	Características usadas
ST	e, cntxt[1-6], CNTXT[1-7], EE[1-3], pos, eMy, eDic, eDisp, cntxtDisp, eLem, eRaíz
STred	e, cntxt[1-6], CNTXT[1-7], pos, eMy, eDic, eDisp, eLema, eRaíz
CNTXr	e, cntxt[1-4], CNTXT[1-5], pos, eMy, eDic, eDisp, eLema, eRaíz
IDL1	e, cntxt[1-4], CNTXT[1-5], pos, eMy
IDL2	e, cntxt[1-4], CNTXT[1-5], pos, eMy, SubStr[1-5]

Cuadro 1: *Conjuntos de características para detección con TiMBL*

Etiquetas	B(%)			I(%)			BIO(%)		
Reconocedor	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$
TMB_{ST}	94.42	95.19	94.81	87.25	85.67	86.45	92.51	92.61	92.56
TMB_{STred}	94.63	94.01	94.32	87.99	85.07	86.50	92.86	91.58	92.22
TMB_{CNTXr}	94.47	95.11	94.79	87.28	85.37	86.31	92.56	92.47	92.51
HMM_{DL}	92.18	93.82	92.99	83.94	81.98	82.95	90.01	90.60	90.31
HMM_{IDL}	92.40	93.99	93.19	83.71	81.00	82.33	90.13	90.46	90.29
VDL	95.31	95.36	95.34	88.02	87.56	87.79	93.34	93.24	93.29
TMB_{IDL1}	94.33	94.91	94.62	87.00	85.29	86.14	92.38	92.30	92.34
TMB_{IDL2}	94.17	95.28	94.72	87.62	85.37	86.48	92.44	92.59	92.51
HMM_{IDL}	92.40	93.99	93.19	83.71	81.00	82.33	90.13	90.46	90.29
VIDL	94.43	95.73	95.07	88.31	86.05	87.17	92.81	93.10	92.96

Cuadro 2: *BIO con corpus completo de entrenamiento*

disminuyen en 0,34 % respecto a ST, consigue mejorar la precisión en la etiqueta I y su tiempo de procesamiento es menor. $CNTXr$ reduce la ventana contextual de 3 a 2, consiguiendo mejores porcentajes con respecto al conjunto $STred$. Por otro lado, creamos dos conjuntos independientes del lenguaje, ya que no hacen uso de recursos dependientes de un idioma concreto, $IDL1$ e $IDL2$. $IDL2$ mejora los resultados globales de $IDL1$, aunque éste último obtiene mejor precisión para la etiqueta B. $IDL1$ mejora al conjunto $STred$ dependiente del lenguaje, mientras que $IDL2$ resulta similar al conjunto $CNTXr$ también dependiente.

En el caso del modelo de HMM las características pueden ser incorporadas mediante la transformación del corpus o de las etiquetas. Estudiamos ambas posibilidades consiguiendo mejores resultados realizando una transformación de las etiquetas. De esta manera creamos HMM_{DL} (dependiente del lenguaje), añadiéndoles a las etiquetas características binarias sobre mayúscula inicial y si la palabra se encuentra en los diccionarios, y HMM_{IDL} (independiente del lenguaje) que incorpora en las etiquetas información sobre el carácter inicial y la palabra entera en mayúsculas. Los resultados se muestran en el Cuadro 2 y aunque HMM obtiene los peores resultados, resulta beneficioso disponer de estos sistemas para complementar el *voting*.

Dividimos los clasificadores en dos grupos: los dependientes del lenguaje y los independientes del lenguaje, realizamos la combinación de dichos clasificadores mediante la estrategia de *voting* explicada en la sección 2.2 obteniendo un 93.29 % en *VDL dependiente del lenguaje* y un 92,96 % en *VIDL independiente del lenguaje*. Esta diferencia de 0.33 % nos indica como un pequeño conjunto de características con atributos independientes del lenguaje puede conseguir resultados similares a los conjuntos dependientes del lenguaje.

5.2. NE-detección con el corpus reducido de aprendizaje

En esta sección comentaremos los resultados obtenidos para la tarea de detección de entidades empleando el corpus de entrenamiento propuesto en la sección 4. Los resultados se muestran en el Cuadro 3, y observándolos denotamos que para casi todos los clasificadores individuales la medida $F_{\beta=1}$ del sistema completo se reduce suavemente, exceptuando al clasificador TBL_{STred} que mejora sus resultados en 0,24 % y HMM_{DL} que los mejora en 0,04 % con respecto a sus análogos con el corpus completo.

El *voting* que se realiza con los clasificadores dependientes del lenguaje obtiene un 0,44 % menos al haber reducido el tamaño del corpus y el *voting* independiente del lenguaje reduce sus resultados en 0,22 %.

Etiquetas	B(%)			I(%)			BIO(%)		
Reconocedor	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$
TMB_{ST}	94.15	95.00	94.57	87.14	85.37	86.25	92.29	92.38	92.34
TMB_{STred}	94.26	95.11	94.68	87.24	85.60	86.41	92.39	92.53	92.46
TMB_{CNTXr}	94.35	94.88	94.62	87.10	85.52	86.30	92.42	92.34	92.38
HMM_{DL}	91.99	93.90	92.93	84.09	82.50	83.29	89.90	90.81	90.35
HMM_{IDL}	92.22	93.90	93.05	83.81	81.22	82.50	90.02	90.46	90.24
VDL	94.72	95.28	95.00	87.39	86.73	87.06	92.75	92.96	92.85
TMB_{IDL1}	94.19	94.38	94.29	87.08	85.37	86.21	92.29	91.93	92.11
TMB_{IDL2}	93.98	95.14	94.55	87.43	85.52	86.47	92.24	92.53	92.38
HMM_{IDL}	92.22	93.90	93.05	83.81	81.22	82.50	90.02	90.46	90.24
VIDL	94.21	95.62	94.91	87.81	85.82	86.80	92.52	92.96	92.74

Cuadro 3: *BIO con corpus reducido de entrenamiento*

No obstante, estos resultados se muestran esperanzadores, ya que consiguen acercarse mucho a los mejores resultados con el corpus de entrenamiento completo.

5.3. NE-clasificación con el corpus completo de entrenamiento

Finalizada la tarea de detección de entidades, realizamos la clasificación de las mismas. Utilizamos las categorías definidas en CoNLL-2002 (PER, ORG, LOC y MISC). Empleamos los resultados obtenidos por la detección de entidades independiente del lenguaje, pues consideramos más relevante utilizar esta aproximación para así poder extrapolarla en un futuro próximo a otras lenguas. En este caso, utilizamos los tres clasificadores (ME, TiMBL y HMM), para ME y TiMBL, realizamos distintos conjuntos de características al igual que hicimos en la detección. Estos conjuntos son mostrados en el Cuadro 4 y utilizan las características definidas en la sección 3.2.

Los resultados obtenidos por cada clasificador con los conjuntos creados, se detallan en el Cuadro 5. Los resultados de aplicar ME con el conjunto Cr, el cual se creó extrayendo de C las características que aportaban mayor información, no han sido mostrados, ya que ME precisa de un mayor número de características para realizar un buen aprendizaje. Por el contrario, TiMBL sí se comporta mejor con el conjunto Cr obteniendo mejores porcentajes para ORG y LOC que con C, además es el clasificador con $F_{\beta=1}$ más alta para la clase LOC con 79.12 %.

A los conjuntos C y Cr se les añadió la característica clx definida en la Figura 3, definidos como Cx y Crx. Con Crx se obtienen resultados más bajos para ORG y LOC, pero mejores para MISC y PER. ME con Cx mejora para PER y LOC, y obtiene para MISC los mejores resultados de entre todos los clasificadores con 58,22 %. TiMBL con Cx también mejora ligeramente los resultados para PER y ORG. De entre todos, HMM resulta el peor clasificador, no obstante HMM lo consideramos un modelo muy rápido y un buen apoyo para el *voting* entre los clasificadores. La estrategia de *voting* fue aplicada con los clasificadores M_C, T_{Cx} y H.

Por último, destacar como la adición o eliminación de características provoca la diversidad de resultados de cada clasificador para cada una de las clases definidas. Sin embargo, todos se comportan bastante bien con las clases PER, ORG y LOC, mientras desarrollan porcentajes más bajos con respecto a la clase MISC. Esto es debido a la dificultad que conlleva a los clasificadores aprender a reconocer una clase tan heterogénea.

5.4. NE-clasificación con el corpus reducido de entrenamiento

Para realizar una valoración del tamaño del corpus de entrenamiento elegido en la sección 4, nos disponemos, al igual que en el caso de la tarea de detección de entidades, a realizar los mismos experimentos para la clasificación con el corpus de entrenamiento reducido. Los resultados se pueden observar en el Cuadro 6. Comparando las medidas $F_{\beta=1}$ con respecto a los clasificadores entrenados con el corpus completo, observamos que para la clase LOC todos mejoran menos HMM, para la clase ORG mejoran todos menos M_C, para MISC solo mejoran sus resultados HMM y M_C y para PER los mejoran T_{Cr} y T_{Crx}. Esto nos demuestra que al reducir el tamaño del corpus han habido relaciones clasificador-

Conjunto	Características usadas
C	e, cntxt[1-6], pos, eP, eL, eO, eM, tP, tL, tO, gP, gL, gO, wP, wL, wO, NoE, f, s
Cr	e, cntxt[1], eP, gP, gL, gO, wP, wL, wO, NoE, f
Cx	e, cntxt[1-6], pos, eP, eL, eO, eM, tP, tL, tO, gP, gL, gO, wP, wL, wO, NoE, f, s, clx
Crx	e, cntxt[1], eP, gP, gL, gO, wP, wL, wO, NoE, f, clx

Cuadro 4: *Conjuntos de características para clasificación*

Etq	LOC(%)			MISC(%)			ORG(%)			PER(%)		
Rcnd	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$
M_C	81.16	74.72	77.81	69.29	49.12	57.49	74.21	84.07	78.83	82.95	88.03	85.41
T_C	75.70	75.28	75.49	55.03	51.47	53.19	75.22	79.79	77.44	84.53	83.27	83.89
M_{Cx}	81.94	74.91	78.27	69.67	50.00	58.22	73.92	84.00	78.64	83.18	88.16	85.60
T_{Cx}	74.84	75.46	75.15	55.88	50.29	52.94	75.88	79.79	77.79	85.42	85.31	85.36
T_{Cr}	80.08	75.65	77.80	57.95	48.24	52.65	77.01	81.36	79.12	79.24	88.30	83.53
T_{Crx}	79.20	75.18	77.14	63.20	50.00	55.83	76.14	81.36	78.66	80.15	88.44	84.09
H	74.85	67.80	71.15	44.66	46.76	45.69	72.06	73.86	72.95	66.11	74.83	70.20
Vot	81.16	75.92	78.46	66.80	49.71	57.00	75.06	83.21	78.93	83.72	89.52	86.52

Cuadro 5: *NE-clasificación con el corpus de entrenamiento completo*

clase que se adaptan mejor para dicha reducción, sin embargo la mayor mejora es de 1,23 % para el clasificador T_{Crx} con la clase PER y el clasificador que más reduce sus resultado es T_C en 0,92 % para la clase PER. Al tratarse de porcentajes bajos podemos afirmar que la reducción del tamaño del corpus ha sido acertada, pues compensa lo suficiente en relación con el tiempo de procesamiento.

6. Comparativa con otros sistemas

Para realizar una valoración más de nuestro sistema, nos atrae la idea de realizar una comparativa con otros sistemas que se encarguen de resolver la tarea de reconocimiento de entidades. Para ello, escogemos los sistemas que participaron en CoNLL-2002. Recordar que en esta comparativa tanto nuestro sistema con el corpus de entrenamiento completo, que nombramos $NERUA$, como el desarrollado con el corpus de entrenamiento reducido $NERUAr$, utilizan la detección de entidades independiente del lenguaje.

En el Cuadro 7 comparamos los resultados totales para nuestros dos sistemas con los participantes en CoNLL 2002. Respecto a f-score nuestro sistema $NERUA$ consigue 1.44 % más que el sistema CY (Cucerzan y Yarowsky, 2002), y obtiene 0.46 % y 2.8 % menos respecto a los sistemas Flo (Florian et al., 2002) y CMP (Carreras, Màrques, y Padró, 2002). Por último, resaltar la escasa diferencia que existe entre $NERUA$, con el corpus de entrenamiento completo, y $NERUAr$ con el corpus de entrenamiento reducido.

Clasif	Prec	Rec.	$F_{\beta=1}$
CMP	81.36	81.40	81.39
Flo	78.70	79.40	79.05
NERUA	78.09	79.10	78.59
NERUAr	77.95	78.95	78.45
CY	78.19	76.14	77.15

Cuadro 7: *Comparativa con otros sistemas*

7. Conclusión y trabajo futuro

El sistema presentado constituye una combinación de diferentes métodos de aprendizaje automático para el desarrollo de la tarea de reconocimiento de entidades para el Español. Proponemos una primera aproximación hacia conseguir un sistema completo de reconocimiento de entidades independiente del lenguaje, realizando una detección de las entidades mediante características propias del texto, sin utilizar recursos ni herramientas externas que dependan de un lenguaje en concreto. Estudiamos los conjuntos de características utilizados por los clasificadores, y así conseguir el mejor conjunto para cada clasificador.

Respecto al corpus de entrenamiento, realizamos exhaustivas pruebas para determinar el tamaño necesario por los clasificadores. Presentamos los resultados obtenidos tanto con el corpus completo como con el corpus reducido que proponemos, pues consideramos crucial saber cuanto corpus es necesario pa-

Etq	LOC(%)			MISC(%)			ORG(%)			PER(%)		
Rcnd	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$
M_C	80.83	75.46	78.05	68.83	50.00	57.92	74.38	83.79	78.80	83.36	87.21	85.24
T_C	74.95	76.20	75.57	55.31	50.59	52.84	75.67	80.43	77.98	84.80	81.22	82.97
M_{Cx}	81.41	75.55	78.37	69.39	50.00	58.12	74.15	83.79	78.67	83.42	87.62	85.47
T_{Cx}	74.75	75.92	75.33	55.88	50.29	52.94	76.19	80.00	78.05	85.71	84.90	85.30
T_{Cr}	80.47	76.01	78.18	55.56	48.53	51.81	77.24	81.21	79.18	79.68	88.03	83.65
T_{Crx}	79.57	75.46	77.46	61.29	50.29	55.25	75.76	81.93	78.72	82.65	88.16	85.32
H	73.53	68.17	70.75	45.66	46.47	46.06	71.80	74.21	72.99	66.79	73.33	69.91
Vot	80.72	76.11	78.35	66.67	50.00	57.14	75.19	82.93	78.87	83.42	88.98	86.11

Cuadro 6: *NE-clasificación con el corpus reducido de entrenamiento*

ra el aprendizaje de los clasificadores, ya que este conocimiento nos ayudará a reducir el tiempo de procesamiento total y hará más cómoda la tarea de etiquetado u obtención de corpus para el entrenamiento. Por último realizamos una comparativa con los sistemas presentados en CoNLL-2002 donde conseguimos un tercer puesto con 78,59 %.

Resumiendo con resultados, el sistema con el corpus completo de entrenamiento, obtiene 92,96 % f-score para la tarea de detección de entidades independiente del lenguaje y 78,59 % f-score para la tarea de clasificación. Mientras que con el tamaño óptimo del corpus de entrenamiento obtenemos 92,74 % f-score para la detección independiente del lenguaje y 78,45 % f-score para la clasificación, ganando en tiempo de procesamiento global del sistema.

Como trabajo futuro pretendemos conseguir por un lado, un sistema completo de reconocimiento de entidades independiente del lenguaje, tanto para la tarea de detección como para la clasificación, y por otro lado obtener sistemas monolingües aplicando diccionarios específicos para entidades, utilizando información morfológica, sintáctica y semántica, incorporando módulos de WSD, etc. Al disponer de ambos sistemas de reconocimiento resultaría una tarea relativamente sencilla adaptar nuestro sistema a otras lenguas como Inglés, Catalán, Italiano, Francés, etc. Un tarea pendiente consistiría en enriquecer nuestro sistema para que realice un reconocimiento expandido de las entidades, por ejemplo para la frase "El presidente de los EEUU"el sistema detectaría *"EEUU"* como una entidad tipo LOC y *"El presidente de los EEUU"* como una entidad tipo PER. Con esta mejora facilitaríamos con nuestro reconocimiento de entidades una mayor comprensión automática del texto.

Bibliografía

Carreras, Xavier, Lluís Màrques, y Lluís Padró. 2002. Named entity extraction using adaboost. En *Proceedings of CoNLL-2002*, páginas 167–170. Taipei, Taiwan.

Cucerzan, Silviu y David Yarowsky. 2002. Language independent ner using a unified model of internal and contextual evidence. En *Proceedings of CoNLL-2002*, páginas 171–174. Taipei, Taiwan.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, y Antal van den Bosch. 2003. TiMBL: Tilburg Memory-Based Learner. Informe Técnico ILK 03-10, Tilburg University.

Florian, Radu, Abe Ittycheriah, Hongyan Jing, y Tong Zhang. 2002. Named entity recognition through classifier combination. En *Proceedings of CoNLL-2002*, páginas 168–171. Taipei, Taiwan.

Rössler, M. 2002. Using markov models for named entity recognition in german newspapers. En *Proceedings of the Workshop on Machine Learning Aproaches in Computational Linguistics*, páginas 29–37. Trento, Italy.

Sang, Tijong Kim. 2002. Introduction to the conll-2002 shared task: Language independent named entity recognition. En *Proceedings of CoNLL-2002*, páginas 155–158.

Schröder, Ingo. 2002. A case study in part-of-speech tagging using the icopost toolkit. Informe Técnico FBI-HH-M-314/02, Department of Computer Science, University of Hamburg.

Suárez, Armando y Manuel Palomar. 2002. A maximum entropy-based word sense disambiguation system. En *COLING 2002*, páginas 960–966.