

Aprendizaje automático para el reconocimiento temporal multilingüe basado en TiMBL*

Marcel Puchol-Blasco Estela Saquete Patricio Martínez-Barco
Dept. de Lenguajes y Sistemas Informáticos (Universidad de Alicante)
Carretera San Vicente s/n 03690 Alicante España
{marcel,stela,patricio}@dlsi.ua.es

Resumen: Este artículo presenta un sistema basado en aprendizaje automático para el reconocimiento de expresiones temporales. El sistema utiliza la aplicación TiMBL, la cuál consiste en un sistema de aprendizaje automático basado en memoria. La portabilidad que presenta este sistema hacia otros idiomas nuevos posee un coste muy reducido, ya que prácticamente no requiere de ningún recurso dependiente del lenguaje (únicamente requiere un tokenizador y un desambiguador léxico categorial, aunque la carencia del POS tagger no repercute mucho en los resultados finales del sistema). Este sistema ha sido evaluado para tres idiomas distintos: inglés, español e italiano. La evaluación realizada presenta resultados satisfactorios para corpus que contienen un gran número de ejemplos, mientras que obtiene resultados bastante pobres en aquellos corpus que contienen pocos ejemplos.

Palabras clave: información temporal, reconocimiento de expresiones temporales, aprendizaje automático

Abstract: This paper presents a Machine Learning-based system for temporal expression recognition. The system uses the TiMBL application, which is a memory-based machine learning system. The portability of the system to other new languages has a very low cost, because it does not need any dependent language resource (only requires a tokenizer and a POS tagger, although the lack in POS tagger does not have enough repercussions on the final system results). This systems has been evaluated on three different languages: English, Spanish and Italian. The evaluation results are quite successful for corpus having a lot of examples; however it obtains very poor results with corpus that have only a few examples.

Keywords: temporal information, temporal expression recognition, machine learning

1. Introducción

El reconocimiento de expresiones temporales cobra cada día más importancia como tarea dentro del campo del Procesamiento del Lenguaje Natural (PLN). La razón de su importancia reside en que se trata de un paso previo a la resolución de expresiones temporales, tarea que puede utilizarse en otros campos del PLN tales como la Búsqueda de Respuestas Temporal, la realización de resúmenes, la ordenación de eventos, etc.

Como en casi todos los aspectos del PLN, existen dos aproximaciones para el reconocimiento de expresiones temporales: los sistemas basados en conocimiento o reglas y los sistemas basados en aprendizaje automático

(AA).

Una de las características más importantes que deben presentar los sistemas actuales de PLN es la facilidad de adaptación del sistema a nuevas lenguas. En este aspecto, los sistemas basados en reglas poseen un gran inconveniente, ya que el conjunto entero de reglas debe reescribirse y adaptarse a la nueva lengua a tratar. Sin embargo, los métodos de AA presentan una gran ventaja en este aspecto, ya que la adaptación a otras lenguas requiere un coste menor que el de sistemas basados en reglas, ya que, en caso de querer adaptar varios sistemas basados en reglas, se deberán de adaptar cada una de las bases de conocimiento de estos sistemas, mientras que si se desean adaptar varios sistemas basados en AA, generando un sólo corpus anotado, suele ser suficiente para adaptarlos todos. No obstante, un inconveniente importante que presentan estos sistemas reside en la necesi-

* Esta investigación ha sido parcialmente financiada bajo los proyectos QALL-ME (FP6-IST-033860), TEXT-MESS (TIN-2006-15265-C06-01) y GV06/028; y bajo la beca de investigación BF-PI06/182.

dad de un corpus anotado con las expresiones temporales en la nueva lengua a tratar, el cual no siempre está disponible.

En anteriores publicaciones hemos tratado el tema de adaptar un sistema de resolución temporal basado en reglas para el español (TERSEO, mirar Saquete, Muñoz, y Martínez-Barco (2005)), partiendo de la base de la traducción de las reglas mediante métodos de traducción automática.

En la línea de mejorar los resultados obtenidos anteriormente (89 % de medida F para el inglés y 79 % de media F para el italiano), y teniendo en cuenta los buenos resultados ofrecidos por los sistemas de AA presentados en diferentes competiciones (tales como el Time Expression Recognition and Normalization Workshop - TERN 2004¹), se ha decidido cambiar la metodología empleada en algunos módulos de TERSEO.

En este artículo presentamos la adaptación del módulo de reconocimiento de expresiones temporales utilizado por TERSEO a métodos de AA. Para tal fin se ha decidido utilizar el sistema de AA TiMBL (Daelemans, Zavrel, y van der Sloot, 2004).

El artículo se estructura de la siguiente manera: en la sección 2 se describe el sistema de aprendizaje automático utilizado, la sección 3 describe el sistema implementado. La siguiente sección define la evaluación del sistema en tres idiomas distintos y compara los resultados con otros sistemas de AA y con el sistema basado en reglas TERSEO. Finalmente, en la sección 5 se presentan las conclusiones y el trabajo futuro que se pretende desarrollar en esta línea de investigación.

2. Sistema de aprendizaje automático

Actualmente los sistemas de aprendizaje automático han tomado mucho auge en el PLN. Debido a eso, muchos sistemas de aprendizaje automático han sido desarrollados, ampliando el abanico de posibilidades a la hora de seleccionar un sistema para un caso en particular.

Un sistema que ha obtenido buenos resultados en aplicaciones destinadas al PLN es TiMBL². Gracias a los buenos resultados ofrecidos por este sistema y a la disposición del API que presenta (gracias a esta API ha

sido posible crear algunas de las características utilizadas para el aprendizaje del sistema), ha sido seleccionado como aplicación de aprendizaje automático para nuestro sistema. Debido a ello, a continuación se presenta un breve resumen de las características de TiMBL.

2.1. TiMBL

TiMBL (Tilburg Memory-based Learning Environment) es una aplicación que implementa algunos algoritmos basados en memoria. Todos estos algoritmos tienen en común que almacenan algún tipo de representación del conjunto de entrenamiento explícitamente en memoria en la fase de entrenamiento. En la fase de evaluación, los nuevos casos se clasifican mediante la extrapolación del caso almacenado más similar.

El aprendizaje basado en memoria (Memory-based learning - MBL, en inglés) se fundamenta en la hipótesis de que el rendimiento en tareas cognitivas se basa en el razonamiento de las bases de interpretación de nuevas situaciones con respecto a situaciones ya almacenadas en experiencias anteriores, más que en la aplicación de reglas mentales abstractas de experiencias anteriores.

Un sistema MBL contiene dos componentes principales:

- Componente de aprendizaje basado en memoria, el cual se encarga de guardar los ejemplos en memoria.
- Componente de interpretación basado en similitud, el cual utiliza como base el resultado del componente de aprendizaje para poder clasificar los ejemplos propuestos. La similitud entre un ejemplo propuesto y los ejemplos almacenados en memoria en la fase de aprendizaje se calcula mediante la distancia métrica $\Delta(X, Y)$ (mirar ecuaciones 1 y 2). Finalmente será el algoritmo IB1 el encargado de asignar la categoría al ejemplo propuesto, seleccionando el más frecuente dentro del conjunto de ejemplos más similares.

¹<http://timex2.mitre.org/tern.html>

²<http://ilk.uvt.nl/timbl/>

$$\Delta(X, Y) = \sum_{i=1}^n \delta(x_i, y_i) \quad (1)$$

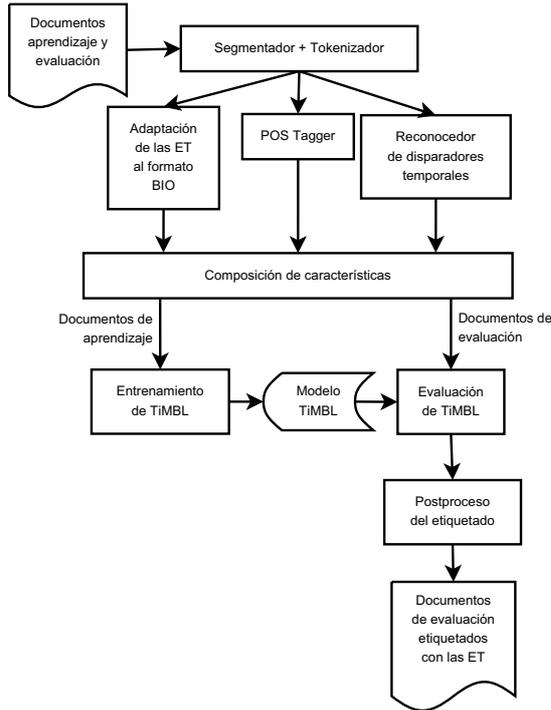


Figura 1: Diagrama del sistema

$$\delta(x_i, y_i) = \begin{cases} \frac{x_i - y_i}{\max_i - \min_i} & \text{si numérico, si no} \\ 0 & \text{si } x_i = y_i \\ 1 & \text{si } x_i \neq y_i \end{cases} \quad (2)$$

3. Descripción del sistema

El sistema propuesto en este artículo se basa en la utilización del sistema de AA TiMBL (comentado en la sección 2.1) para aprender sobre el conjunto de ejemplos generados para la fase de entrenamiento a partir de las distintas características seleccionadas y etiquetar, posteriormente, el conjunto de ejemplos generados para la fase de evaluación.

Para poder generar los ejemplos de entrenamiento y de evaluación se ha seguido la metodología presentada en la figura 1.

Los pasos seguidos para el tratamiento de los documentos son:

1. Segmentación del documento en oraciones.
2. Tokenización de los elementos de la oración.
3. Extracción del POS de cada token.
4. Adaptación de las expresiones temporales al formato BIO.

5. Reconocimiento de los posibles tokens que sean disparadores temporales.
6. Composición de las características de entrenamiento si se trata de un documento destinado a la fase de entrenamiento o composición de las características de evaluación si se trata de un documento destinado a la fase de evaluación.
7. Clasificación de los ejemplos mediante TiMBL.
8. Postprocesamiento de la salida de TiMBL.

Tomemos como ejemplo la siguiente oración:

La alarma sonó <TIMEX2> cuatro horas antes de la explosión </TIMEX2>.

La oración se tokeniza, se utiliza un PoS-tagger³ para obtener la categoría léxica de cada token y se adaptan las expresiones temporales al formato BIO (Begin - inicio de la expresión temporal; Inside - dentro de la expresión temporal; y Outside - fuera de la expresión temporal), generándose una distribución vertical como la que se muestra a continuación:

La	O	DA0FS0
alarma	O	NCFS000
sonó	O	VMIS3S0
cuatro	B	Z
horas	I	NCFP000
antes	I	RG
de	I	SPS00
la	I	DA0FS0
explosión	I	NCFS000
.	O	Fp

A continuación se realiza el reconocimiento de los disparadores temporales, en el cual se analiza token a token si pertenece o no a la ontología de disparadores temporales siguiente:

- Día de la semana: lunes, martes, miércoles...
- Meses del año: enero (ene.), febrero (feb.), marzo (mar.)...
- Estaciones del año: primavera, otoño, invierno o verano.
- Festividades: Navidad, Epifanía, Adviento, Halloween...

³Etiquetador léxico categorial

- Palabras temporales: ayer, anteayer, hoy, mañana, tarde, noche, anteanoche, tiempo, presente, pasado, futuro, hora, minuto, segundo...
- Posibles preposiciones temporales: durante, entre, hasta...
- Posibles adverbios temporales: antes, después...
- Números
- Fechas simples: dd/mm/aaaa

El siguiente paso a realizar consiste en generar los ejemplos necesarios para que el sistema de AA aprenda de ellos. Para ello es necesario extraer una serie de características de las oraciones. Las características que han sido consideradas en este sistema pueden agruparse en:

- Características relacionadas con el token (TOK): TOK₀, BGR(TOK₋₁ TOK₀), BGR(TOK₀ TOK₁), BGR(TOK₋₂ TOK₋₁), BGR(TOK₁ TOK₂), SUF(TOK)₂, SUF(TOK)₃, PREF(TOK)₂, PREF(TOK)₃.
- Características relacionadas con los disparadores (DISP): BGR(DISP₋₁ DISP₀), BGR(DISP₀ DISP₁).
- Características relacionadas con los ejemplos ya etiquetados de la oración (ETIQ): ETIQ₋₁, BGR(ETIQ₋₂ ETIQ₋₁), ETIQ₁, BGR(ETIQ₁ ETIQ₂)⁴.
- Características relacionadas con el POS: POS₁.

NOTAS: TOK (token), DISP (disparador), ETIQ (elemento ya etiquetado), BGR (bigrama).

Sin embargo, éstas no han sido las únicas que inicialmente se consideraron en el sistema. Las siguientes características fueron consideradas como una posible mejora al sistema, pero al obtener peores resultados, se descartaron del sistema:

- Características relacionadas con el token (TOK): BGR(TOK₋₃ TOK₋₂), BGR(TOK₂ TOK₃).

⁴Posteriormente se verá el tratamiento de este tipo de características

- Características relacionadas con los disparadores (DISP): BGR(TOK₋₂ TOK₋₁), BGR(TOK₁ TOK₂).
- Características relacionadas con ejemplos ya etiquetados en la oración (ETIQ₁): BGR(ETIQ₂ ETIQ₃), BGR(ETIQ₋₂ ETIQ₋₃).
- Características relacionadas con ejemplos ya etiquetados en la oración (ETIQ₂):
$$\forall_{x \in [ET_{ini}, 0]} \begin{cases} DISP_x & : \exists DISP_x \\ TOK_x & : \text{otro caso} \end{cases}$$
- Características relacionadas con ejemplos ya etiquetados en la oración (ETIQ₃): $\forall_{x \in [ET_{ini}, 0]} TOK_x si TOK_x \notin STOPWORDS$.

Acrónimos utilizados: TOK (token), DISP (disparador), ETIQ (elemento ya etiquetado), BGR (bigrama). Posiciones utilizadas: 0 (posición actual), -x (x posiciones anteriores), x (x posiciones posteriores), ET_{ini} (posición de inicio de la expresión temporal actual).

Es importante remarcar que las características relacionadas con ejemplos ya etiquetados reciben un tratamiento diferente en la fase de entrenamiento y en la fase de evaluación. En la fase de entrenamiento sí se posee esta información, mientras que en la fase de evaluación ha sido necesario realizar una serie de cambios al funcionamiento normal de TiMBL para poder tratar este tipo de características. El siguiente algoritmo explica el funcionamiento seguido para tratar este tipo de característica:

Primera pasada - Descendente

Para cada ejemplo descendentemente

@num = CLASE[POS-num]

CAR[#num] = NADA

Clasificar

Guardar CA

Fin Para

Segunda pasada - Ascendente

Para cada ejemplo descendentemente

@num = CLASE[POS-num]

#num = CLASE[POS+num]

Clasificar

Si CA ≠ CAA entonces

Tercera pasada - descendente

POS₃ = POS + 1

Hacer

Tomar ejemplo

```

@num = CLASE[POS3-num]
#num = CLASE[POS3+num]
Clasificar
POS3++
Mientras CA ≠ CAA
Fin Para

```

NOTAS: CA (clase asignada), CAA (clase asignada anteriormente), @ (clase anterior), # (clase posterior), CAR (característica).

En la figura 2 puede verse un ejemplo de una traza realizada para este algoritmo.

Una vez etiquetados todos los ejemplos, se realizará un postproceso muy simple de coherencia de las etiquetas de salida del sistema de AA. Este postproceso se basará en comprobar si existe alguna clasificación con etiqueta I que posea en la posición anterior la etiqueta O y modificará esa etiqueta I por la etiqueta B.

Una vez realizado todo este proceso, los documentos de evaluación estarán etiquetados con las expresiones temporales.

4. Resultados experimentales

Ha decidido probarse el sistema en tres idiomas distintos: inglés, español e italiano. Para cada uno de estos idiomas se ha seleccionado un corpus etiquetado mediante etiquetas TIMEX2, los cuales serán detallados a continuación. Debido a que la finalidad de evaluar este sistema no se basa en comparaciones con los sistemas ya existentes, sino que se intentan conseguir los mejores resultados posibles, se ha utilizado el método de evaluación 3-fold cross validation. El sistema de evaluación utilizado para medir las prestaciones del sistema es el proporcionado oficialmente en el TERN, el cual se basa en un script desarrollado por el MITRE para la evaluación de sistemas. Los resultados son mostrados utilizando valores de precisión y cobertura con la métrica $F_{\beta=1}$. Finalmente se muestran las conclusiones derivadas de los resultados obtenidos.

4.1. Corpora utilizado

El corpus utilizado para el inglés es el proporcionado en el TERN 2004⁵. Este corpus está formado por documentos de noticias extraído de los periódicos, transmisiones de noticias y agencias de noticias. Para el proceso

⁵<http://timex2.mitre.org/tern.html>

de evaluación realizado, los corpus de entrenamiento y evaluación se han unido.

El corpus utilizado para el español se basa en una serie de documentos extraídos de periódicos digitales en castellano utilizados en anteriores evaluaciones del sistema TERSEO.

El corpus utilizado para el italiano se denomina I-CAB. Este corpus fue creado como parte del proyecto ONTOTEXT⁶. Este corpus está formado por documentos de noticias extraídos del periódico local L'Adige. La anotación se ha llevado a cabo siguiendo los estándares del programa ACE (Automatic Content Extraction⁷) para la tarea de Reconocimiento y Normalización de Expresiones Temporales (Ferro et al., 2005).

Las características más importantes de estos tres corpus pueden verse en la tabla 1.

Idioma	DOCS	TOK	ET
Inglés	511	196.473	4.728
Español	100	39.719	431
Italiano	528	204.185	4.548

Cuadro 1: Información sobre los corpora utilizados para evaluar el sistema

4.2. Proceso de evaluación

Como ha sido comentado anteriormente, se generaron una serie de características iniciales sobre las que se realizó una selección para obtener las mejores. Esta selección se realizó según el método de Moreda y Palomar (2005), obteniendo aquellas que pondrían finalmente el sistema. Para aquellas características relacionadas con la información léxico-categorial, se utilizó la herramienta FreeLing (Atserias et al., 2006).

La evaluación de los resultados de reconocimiento de expresiones temporales para los distintos idiomas, teniendo en cuenta la medida obtenida por el scorer del TERN como TIMEX2⁸ y las características seleccionadas finalmente en el sistema, se muestra en la tabla 2, mientras que la medida obtenida por el scorer del TERN como TIMEX2:TEXT⁹, con las mismas características, se muestra en la tabla 3.

⁶<http://tcc.itc.it/projects/ontotext>

⁷<http://www.nist.gov/speech/tests/ace>

⁸medida de comprobación de la detección de expresiones temporales

⁹medida de la extensión de la ET (comprobación de los límites de las ET)

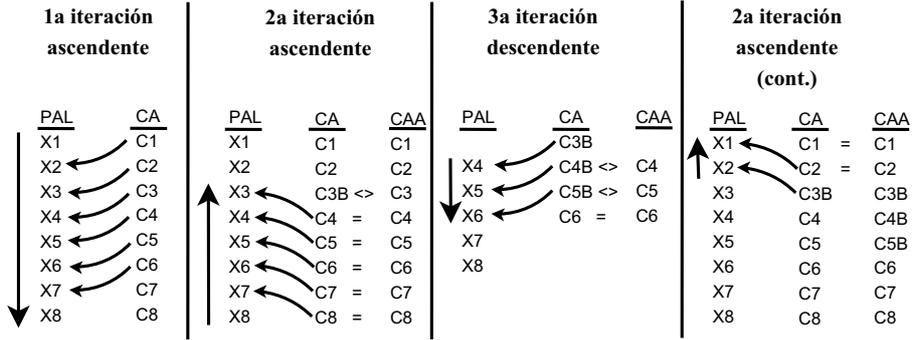


Figura 2: Ejemplo de traza del algoritmo de características relacionadas con ejemplos ya etiquetados

Características	Inglés			Castellano			Italiano		
	P	R	F	P	R	F	P	R	F
TOK	0.654	0.839	0.735	0.503	0.683	0.579	0.630	0.755	0.687
TOK+DISP	0.713	0.872	0.784	0.541	0.795	0.642	0.661	0.792	0.721
TOK+DISP+ETIQ	0.861	0.823	0.841	0.742	0.673	0.705	0.791	0.740	0.765
TOK+DISP+ETIQ+POS	0.871	0.833	0.851	0.744	0.708	0.725	0.784	0.748	0.765

Cuadro 2: Resultados del sistema para TIMEX2

Como puede observarse, se ha realizado una evaluación incremental del tipo de características ejecutadas para demostrar el avance de las mismas. Las medidas mostradas en las tablas corresponden a: P (Precisión), R (Recall - Cobertura), F (medida F).

Como puede observarse, los resultados obtenidos en los idiomas en los que los corpora poseen más ejemplos para el aprendizaje del sistema, obtienen mejores resultados.

Otro factor importante es la incorporación de las clasificaciones realizadas anteriormente, junto con el algoritmo de múltiples pasadas realizado para conocer, tanto las clasificaciones anteriores, como las posteriores. Como puede observarse, este tipo de características pueden mejorar los resultados del sistema más de un 10% de precisión. Sin embargo, la incorporación de información del POS al sistema mejora únicamente en un 1%. Este efecto plantea si es realmente necesario incorporar un recurso dependiente del lenguaje (el POS tagger¹⁰) al sistema para obtener una mejora tan ínfima.

4.3. Comparación con otros sistemas

Debido al sistema de evaluación utilizado (3-fold cross validation), no puede realizarse una comparación directa con otros siste-

¹⁰Desambiguador léxico categorial

mas de aprendizaje automático, ya que otros sistemas utilizan distintos tipos de métodos de evaluación. Sin embargo, comparando este sistema con sistemas como el de Hacıoglu, Chen, y Douglas (2005), podemos apreciar que el sistema presentado en este artículo ofrece menores resultados de precisión y cobertura. Sin embargo, al analizar el método utilizado para la evaluación se observa que los resultados presentados en este artículo son más contundentes, ya que consideramos que el 3-fold cross validation proporciona unos resultados más fiables que los empleados en este artículo. Además, también se debe tener en cuenta el tipo de requerimientos que posee un sistema y el otro. Mientras que este sistema sólo necesita de un segmentador, un tokenizador y un POS tagger, el otro sistema necesita, además de lo mismo que este, de un parser¹¹ y un chunker¹².

Si comparamos los resultados obtenidos en este artículo con los obtenidos anteriormente en TERSEO (Saquete et al., 2006) apreciamos que los resultados para el inglés son bastante parecidos, mientras que para el italiano baja un poco la precisión. Sin embargo, al compararlo con el idioma origen de TERSEO

¹¹Sistema que realiza un análisis sintáctico total de la oración

¹²Sistema que realiza un análisis sintáctico parcial de la oración

Características	Inglés			Castellano			Italiano		
	P	R	F	P	R	F	P	R	F
TOK	0.563	0.722	0.633	0.360	0.487	0.413	0.524	0.628	0.571
TOK+DISP	0.596	0.731	0.657	0.387	0.572	0.462	0.546	0.655	0.596
TOK+DISP+ETIQ	0.756	0.723	0.739	0.585	0.531	0.556	0.667	0.625	0.646
TOK+DISP+ETIQ+POS	0.766	0.733	0.749	0.582	0.553	0.567	0.664	0.633	0.648

Cuadro 3: Resultados del sistema para TIMEX2:TEXT

(mirar (Saquete, Muñoz, y Martínez-Barco, 2005)), el español, los resultados se inclinan favorablemente hacia TERSEO, el cual obtiene un 80 % de precisión frente al 72 % obtenido por este sistema. Sin embargo, si tenemos en cuenta los resultados obtenidos por este sistema sin la necesidad de ningún recurso dependiente del lenguaje (70 % de medida F) y que TERSEO requiere de recursos dependientes del lenguaje para su funcionamiento (TERSEO necesita un POS tagger), los resultados ofrecidos por este sistema son bastante satisfactorios en este aspecto. Además, el coste asociado a la adaptación de TERSEO a otros lenguajes distintos del español es mucho más grande que el asociado a este sistema.

5. Conclusiones y trabajo futuro

Se ha presentado un sistema basado en aprendizaje automático basado en TiMBL que posee un bajo coste de adaptabilidad a otros idiomas, siempre y cuando exista un corpus etiquetado con ETs en la lengua que se desee tratar. Este sistema ha sido probado en tres idiomas distintos: inglés, español e italiano. Los resultados obtenidos para los idiomas que poseen un corpus con muchos ejemplos en los que basarse el sistema de aprendizaje automático ofrecen resultados satisfactorios (en inglés, un 85 % para la evaluación TIMEX2 y un 75 % para la evaluación TIMEX2:TEXT, mientras que en italiano, un 76 % para la evaluación TIMEX2 y un 65 % para la evaluación TIMEX2:TEXT). Sin embargo, se ha comprobado como en corpus con pocos ejemplos de los que aprender, se obtienen unos resultados bastante pobres (en español, un 72 % para la evaluación TIMEX2 y 57 % para la evaluación TIMEX2:TEXT).

Como puede comprobarse, estos resultados son favorables y suficientes para la incorporación de este sistema en el módulo de TERSEO de reconocimiento de expresiones temporales, pese a que el módulo de TER-

SEO ofrezca mejores resultados. Es necesario tener en cuenta que TERSEO depende de recursos lingüísticos dependientes del lenguaje, muchos de los cuales no existen en determinados idiomas, mientras que en este sistema estos recursos son prescindibles.

Como trabajo futuro, quieren realizarse pruebas con otra serie de características que requieran de una mejor comprensión del texto. En concreto se desea utilizar información sintáctica y semántica. Además, este sistema quiere incorporarse completamente como módulo de reconocimiento de expresiones temporales de TERSEO. Además, siguiendo una estrategia similar, quieren realizarse pruebas de adaptación a la tecnología de aprendizaje automático en otros módulos dependientes del idioma de TERSEO. Finalmente se desea evaluar la combinación completa de TERSEO con los módulos de dependientes del sistema basados en aprendizaje automático y los módulos independientes del sistema, los cuales están basados en reglas, comprobando la precisión final de TERSEO tanto en reconocimiento como en resolución de expresiones temporales.

Bibliografía

- Atserias, J., B. Casas, E. Comelles, M. González, L. Padró, y M. Padró. 2006. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. En *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, páginas 48–55.
- Daelemans, W., J. Zavrel, y K. van der Sloot. 2004. TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide. Ilk research group technical report series, Tilburg. 60 pages.
- Ferro, L., L. Gerber, I. Mani, B. Sundheim, y G. Wilson. 2005. Tides.2005 standard for the annotation of temporal expressions. Informe técnico, MITRE.

- Hacioglu, Kadri, Ying Chen, y Benjamin Douglas. 2005. Automatic time expression labeling for english and chinese text. En Alexander F. Gelbukh, editor, *CICLing*, volumen 3406 de *Lecture Notes in Computer Science*, páginas 548–559. Springer.
- Moreda, P. y M. Palomar. 2005. Selecting Features for Semantic Roles in QA Systems. En *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, páginas 333–339, Borovets, Bulgaria, Septiembre.
- Saquete, E., R. Muñoz, y P. Martínez-Barco. 2005. Event ordering using terseo system. *Data and Knowledge Engineering Journal*, página (To be published).
- Saquete, Estela, Óscar Ferrández, Patricio Martínez-Barco, y Rafael Muñoz. 2006. Reconocimiento temporal para el italiano combinando técnicas de aprendizaje automático y adquisición automática de conocimiento. En *Proceedings of the 22nd International Conference of the Spanish Society for the Natural Language Processing (SEPLN)*.