Procesamiento del Lenguaje Natural
ISSN: 1135-5948
secretaria.sepln@ujaen.es
Sociedad Española para el Procesamiento del Lenguaje Natural
España

Enríquez, Fernando; Troyano, José A.; Cruz, Fermín; Ortega, F. Javier

Generación semiautomática de recursos

Procesamiento del Lenguaje Natural, núm. 39, 2007, pp. 173-180
Sociedad Española para el Procesamiento del Lenguaje Natural
Jaén, España

Disponible en: http://www.redalyc.org/articulo.oa?id=515751739021

Cómo citar el artículo
Número completo
Más información del artículo
Página de la revista en redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Procesamiento del Lenguaje Natural, nº39 (2007), pp. 173-180 recibido 18-05-2007; aceptado 22-06-2007

Generación semiautomática de recursos *

Fernando Enríquez, José A. Troyano, Fermín Cruz y **F. Javier Ortega**
Dep. de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Avda. Reina Mercedes s/n
41012 Sevilla
fenros@us.es

Resumen: Los resultados de muchos algoritmos que se aplican en tareas de procesamiento del lenguaje natural dependen de la disponibilidad de grandes recursos lingüísticos, de los que extraen el conocimiento necesario para desempeñar su trabajo. La existencia de estos recursos determina por tanto la calidad de los resultados, el rendimiento general del sistema y en ocasiones, ambas cosas. Vamos a mostrar diversos aspectos que hacen referencia al esfuerzo necesario para la creación de estos recursos, y que por lo tanto justifican los intentos de desarrollar métodos que alivien esta tarea, así como diversas propuestas que se han mostrado para solventar esta cuestión. Estas propuestas pueden considerarse alternativas al problema que queremos solucionar y lo afrontan de muy diferentes maneras, algunas de las cuales quizás podamos adaptar a nuestras propias implementaciones en un futuro próximo.
Palabras clave: Generación de recursos, aprendizaje automático, combinación de sistemas

Abstract: The results of many algorithms that are applied to natural language processing tasks depend on the availability of large linguistic resources from which they obtain the required knowledge to do their work. The existence of these resources determines the quality of the results, the general performance of the system and frequently both things. We are going to show some aspects that refer to the effort needed in the creation of these resources, and thus justify the attempts to develop methods that lighten this task, and also some proposals that have been made to solve this problem. These proposals can be considered alternatives to the problem we want to solve and they face it in very different manners, some of which could be adapted in our own implementations in a near future.
Keywords: Resource generation, machine learning, system combination

1. Introducción

Sin duda alguna el mayor problema que surge a la hora de afrontar la creación de recursos lingüísticos es el esfuerzo que se requiere para obtener resultados de suficiente envergadura como para que les sean útiles a los algoritmos que los necesitan. Generalmente, un algoritmo de aprendizaje supervisado que hace uso de un corpus etiquetado para una determinada tarea, exige un número muy alto de palabras o frases etiquetadas para ofrecer resultados que puedan ser considerados de calidad aunque esto dependerá del algoritmo en cuestión y de la tarea que se esté afrontando.

Si nos centramos en una tarea amplia-

* Parcialmente financiado por el Ministerio de Educación y Ciencia (TIN2004-07246-C03-03).

ISSN: 1135-5948 © 2007 Sociedad Española para el Procesamiento del Lenguaje Natural

mente conocida dentro del procesamiento del lenguaje natural, como es la desambiguación de significados, podemos hacernos una idea de este esfuerzo que estamos comentando. Se trata de una tarea que afronta el problema de seleccionar el significado de una palabra en un texto de entre todos los significados que posee. La ambigüedad es muy común aunque los humanos estamos tan acostumbrados a ella y tenemos tal capacidad de resolverla basándonos en el contexto de las palabras, que casi pasa desapercibida ante nuestros ojos. Para esta tarea se han desarrollado múltiples algoritmos con muy buenos resultados, aunque la disponibilidad de corpus etiquetados sigue constituyendo un problema. En (Ng, 1997) se realizó un estudio que asegura que para obtener una precisión buena se necesitan al menos 500 ejemplos por cada una de las palabras ambiguas a tratar (esta es una cifra que representa la media ya que hay diferencias considerables de una palabra a otra). A un ritmo de un ejemplo etiquetado por minuto y considerando la existencia de unas 20000 palabras ambiguas en el vocabulario inglés común, esto nos conduciría a unas 160000 horas de etiquetado, que resultarían en nada más y nada menos que 80 años de dedicación exclusiva para una persona que lleve a cabo esta tarea de etiquetado. Si además le añadimos el hecho de que las tareas de etiquetado suelen ser llevadas a cabo por lingüistas entrenados o expertos, no cabe duda de que se trata de un proceso realmente caro y generalmente prohibitivo en la inmensa mayoría de los casos.

Todo esto supone una limitación y termina por reducir el número de ejemplos disponibles, afectando a la tarea en general y posiblemente al desarrollo de nuevas vías de investigación que puedan aportar mejoras en los resultados. De ahí que este sea el punto de partida de una linea de trabajo futuro que deseamos recorrer y de la que intentaremos extraer soluciones satisfactorias a este problema.

A lo largo de los sucesivos capítulos veremos algunas técnicas empleadas para crear recursos lingüísticos, comenzando en el capítulo 2 con un algoritmo que emplea consultas en buscadores web. En el capítulo 3 comentaremos las técnicas de crowdsourcing, cuyo uso se está extendiendo con rapidez, mientras que en los capítulos 4 y 5 comentaremos métodos de combinación e importación de recursos respectivamente. En el capítulo 6 veremos las técnicas de bootstrapping para finalizar con un capítulo dedicado a las conclusiones.

2. *Empleando Búsquedas en la Web*

Una de las vías que han surgido para intentar paliar los efectos del enorme esfuerzo requerido para la creación de recursos, es el uso de la Web. El contenido de la Web puede ser considerado un enorme corpus que puede ser explotado para diversas tareas, si bien presenta una estructura y unos contenidos tan heterogéneos que no siempre se sabe muy bien como sacarle partido a toda la información que posee.

En (Mihalcea, 2002) podemos apreciar un magnífico ejemplo de cómo se puede hacer uso de la Web para obtener recursos lingüísticos a través de los sistemas de búsquedas que tenemos a nuestra disposición. La tarea que se afronta en este trabajo es la desambiguación de significados y el sistema propuesto hace uso de diversos recursos disponibles como el corpus SemCor (Miller, 1993) y la base de datos léxica WordNet (Miller, 1995). El algoritmo se resume en la figura 1.

Las semillas están formadas por múltiples unidades de palabras que contienen una palabra ambigua, de forma que la expresión por sí misma supone una restricción para el posible significado de la palabra en la que recae el interés.

En este algoritmo se emplea un método para, utilizando WordNet, construir consultas que contengan sinónimos o definiciones del significado de las palabras de interés y mediante los motores de búsqueda disponibles en Internet, realizar dichas consultas para obtener textos relacionados con esas definiciones. En WordNet se buscan en primer lugar sinónimos que sean monosémicos, y si no existen, se buscan definiciones de la palabra. Al hacer la búsqueda, se seleccionan las oraciones que contengan la definición o el sinónimo y se sustituyen por la palabra original, obteniéndose un ejemplo de uso de dicha palabra con su significado.

Una vez tenemos las expresiones encontradas tras explorar la web haciendo uso de las semillas, se aplica un algoritmo iterativo de desambiguación mediante varios procedimientos cuyas claves se resumen en:

1. Crear un conjunto de semillas, compuestas por:
  1.1 Ejemplos de SemCor.
  1.2 Ejemplos de WordNet.
  1.3 Ejemplos etiquetados creados mediante búsquedas en la web de sinónimos monosémicos o definiciones de la palabra.
  1.4 Ejemplos adicionales etiquetados manualmente (si están disponibles).

2. Realizar búsquedas en la Web utilizando las expresiones de las semillas.

3. Desambiguar las palabras en un contexto cercano al texto que rodea las expresiones de las semillas. Agregar los ejemplos formados con las palabras desambiguadas al conjunto de las semillas.

4. Volver al paso 2.

Figura 1: Algoritmo de búsquedas en la web.

1. Localizar las entidades, como nombres de personas, lugares y organizaciones, y marcar su significado.

2. Localizar las palabras monosémicas y marcar su significado.

3. Para cada palabra se forman pares con la palabra dada y la anterior y posterior. Si en el corpus *SemCor* aparecen dichos pares suficientes veces (superior a un umbral preestablecido) y siempre con el mismo significado, se le asigna dicho significado a la palabra.

4. Para los sustantivos se crea un contexto, conteniendo los sustantivos que suelen aparecer cerca por cada significado posible. Luego se compara con el contexto actual del sustantivo y se escoge el significado más parecido.

5. Se buscan conexiones semánticas entre palabras, por lo que, si una palabra tiene un significado que la convierte en sinónima de otra ya desambiguada, se le asigna dicho significado. También se estudian relaciones de hiperonimia e hiponimia y se buscan conexiones entre palabras estando ambas sin desambiguar.

Los experimentos realizados para medir la calidad de los corpus que se obtienen mediante este algoritmo, demuestran que se obtienen resultados comparables a los adquiridos a través del uso de corpus etiquetados manualmente. Concretamente, los autores hicieron experimentos con diversas herramientas de etiquetado semántico, utilizando un corpus etiquetado manualmente y por otro lado, el corpus obtenido automáticamente mediante este algoritmo. La precisión alcanzada cuando se usaba el corpus automático era a veces incluso mejor que la obtenida con las mismas herramientas pero utilizando el corpus manual.

3. El Crowdsourcing

El crowdsourcing es un término acuñado recientemente y que constituye un paso adelante tras el outsourcing. Este último está basado en la delegación de ciertas tareas en determinadas entidades externas para ahorrar costes y simplificar el proceso de desarrollo en un proyecto (generalmente las empresas han estado fijando las miradas en India o China). Las nuevas posibilidades de ahorro en este entorno es posible que se encuentren en el trabajo disperso y anónimo de multitud de internautas que desarrollan tareas de mayor o menor valor para una organización que sepa llamar su atención de alguna de entre tantas formas posibles. Esta forma de recopilar el esfuerzo y orientarlo hacia la consecución de algún objetivo relacionado con el desarrollo de alguna tarea en concreto se denomina *crowdsourcing* [1].

El precursor de este término es Jeff Howe, quién en (Howe, 2006) comenta varios ejemplos en los que se ha aplicado esta forma de trabajo. En dicho artículo comienza comentando un caso particular referente a un fotógrafo profesional que pierde un cliente al descubrir este que puede comprar fotos a través de iStockPhoto a un precio mucho menor (el cliente solo buscaba fotos de gente enferma para un trabajo que estaba realizando). En este portal se publican un numero muy grande de fotos realizadas por amateurs y que son muy útiles en muchos casos sin necesidad de pagar el alto precio

[1] Del inglés 'crowd' que significa multitud y 'source' que significa fuente

que cobraría un profesional al que le encargase el trabajo de forma directa. Es un ejemplo más en el que el trabajo de miles de personas puede ser aprovechado cambiando un escenario empresarial que parecía en principio inquebrantable. De esta forma cada participante puede publicar todo tipo de fotos cobrando muy poco por cada una pero con la capacidad de ponerlas al alcance de cualquiera que esté conectado a Internet. Esto lleva al autor a decir:

> Welcome to the age of the crowd. Just as distributed computing projects like UC Berkeley's SETI@home have tapped the unused processing power of millions of individual computers, so distributed labor networks are using the Internet to exploit the spare processing power of millions of human brains.

En la misma línea de este ejemplo que acabamos de comentar, hallamos multitud de proyectos, sistemas y aplicaciones que intentan sacar partido de todo este potencial, por ejemplo, la wikipedia, una enciclopedia que se extiende rápidamente entre las preferencias de los usuarios de Internet, y que está hecha mediante la contribución anónima de todos los que quieran aportar su grano de arena a esta recopilación de conocimiento. También lo vemos en los programas de televisión que se basan estrictamente en mostrar el material creado por los propios telespectadores (emitiendo sus videos caseros, composiciones musicales, etc) y que obtienen en muchos casos cifras de audiencia espectaculares sin apenas suponerle ningún coste a la cadena. Otros ejemplos pueden ser, el proyecto InnoCentive, a través del cuál se publican problemas de cierta dificultad técnica o científica que le surgen a todo tipo de empresas, de forma que cualquiera puede intentar darle solución (recibiendo grandes recompensas económicas) o el Turco Mecánico de Amazon, a través del cuál todo el mundo puede cobrar una pequeña cantidad de dinero por realizar tareas muy simples sin necesidad de una gran preparación previa.

La iniciativa 'Open Mind' (Stork, 1999) es el resultado de aplicar esta idea a la generación de recursos lingüísticos. La idea básica es utilizar la información y el conocimiento que se puede obtener a partir de los millones de usuarios de Internet con el objetivo de crear aplicaciones más inteligentes. Dentro de esta iniciativa se encuentran diversos proyectos relacionados con el lenguaje natural como Open Mind Word Expert (Mihalcea, 2003), centrado en la desambiguación de significados (generando corpus anotados semánticamente por los usuarios) y Open Mind Common Sense (Singh, 2002) que se centra en la adquisición del sentido común para generar un corpus textual.

4. La Combinación de Recursos

Otra estrategia que podemos encontrar en la bibliografía para generar corpus es la combinación de recursos ya existentes, de manera que se enriquezcan unos con otros aumentando su valor al ser considerados de forma global. Un ejemplo muy clarificador lo podemos encontrar en (Shi, 2005), donde se combinan FrameNet, VerbNet y WordNet. Vamos a comentar brevemente el contenido de estos recursos para luego comprender cómo se combinan creando un recurso unificado.

- La primera pieza de este puzzle parte de WordNet. Es una gran base de datos léxica con mucha información sobre palabras y conceptos. Este es el recurso utilizado para identificar características semánticas superficiales que pueden asociarse a unidades léxicas. En WordNet se cubren la gran mayoría de nombres, verbos, adjetivos y adverbios del inglés. Las palabras se organizan en conjuntos de sinónimos (llamados 'synsets') que representan conceptos.
- FrameNet por su parte es un recurso que contiene información sobre diferentes situaciones, llamadas 'frames'. Cada frase etiquetada en FrameNet representa una posible construcción sintáctica para los roles semánticos asociados con un frame para una determinada palabra. Solemos referirnos al conocimiento que aporta WordNet como conocimiento a nivel de palabra (word-level knowledge), mientras que FrameNet y VerbNet hacen referencia al conocimiento a nivel de frase (sentence-level knowledge).
- Y finalmente Verbnet es un recurso léxico de verbos basado en las clases de verbos de Levin, y que también aporta restricciones selectivas asociadas a los

roles semánticos. Identificando la clase de VerbNet que se corresponde con un frame de FrameNet, se pueden analizar sintácticamente frases que incluyen verbos que no están cubiertos aún por FrameNet. Se puede hacer esto gracias a que existe una relación transitiva entre las clases de VerbNet (los verbos que pertenecen a la misma clase en VerbNet tienen una alta probabilidad de compartir el mismo frame en FrameNet, y por lo tanto se pueden analizar semánticamente aunque no aparezcan explícitamente en FrameNet).

Dados estos tres recursos, se pueden combinar de manera que se pueda trabajar con todos ellos a la vez, en lugar de estar obligados a elegir sólo uno renunciando a la información que aportan los otros. Las características que permiten llevar a cabo esta unión son las siguientes:

- FrameNet no define explícitamente restricciones de selección para los roles semánticos. Además, la construcción de FrameNet requirió de un gran esfuerzo humano por lo que la cobertura y escalabilidad se han visto seriamente afectadas.
- VerbNet sin embargo tiene mucha mejor cobertura y define relaciones sintacticosemánticas de una manera más explícita. VerbNet etiqueta roles temáticos y proporciona restricciones de selección para los argumentos de los marcos sintácticos.
- WordNet por su parte cubre casi al completo todos los verbos del inglés y aporta una gran información sobre las relaciones semánticas entre los sentidos de los verbos. De todas formas, la construcción de WordNet está basada en el significado de los verbos y no incluye el comportamiento sintáctico o semántico de los mismos (como pueden ser las estructuras de tipo predicado-argumento).

Una vez analizado el contenido de estos tres recursos, la combinación de la información codificada en cada uno de ellos pasa por:

- Aumentar la semántica de los marcos con las clases de VerbNet etiquetando los marcos y los roles semánticos de FrameNet con las entradas de VerbNet y sus argumentos correspondientes.
- También se extiende la cobertura de los verbos de FrameNet haciendo uso de las clases de VerbNet y las relaciones de sinonimia e hiponimia de los verbos de WordNet.
- Además, se identifican las conexiones explícitas entre los roles semánticos y las clases semánticas, codificando restricciones de selección para los roles semánticos mediante la jerarquía de nombres de WordNet.

La construcción de recursos lingüísticos requiere un gran esfuerzo humano y cada recurso está pensado para solucionar un determinado tipo de problemas, mostrando virtudes en ciertos aspectos y desventajas en otros. De esta forma, la combinación de estos recursos puede dar lugar a una base de conocimiento más extensa y más rica. En (Shi, 2005) hemos visto como se mejora la cobertura de FrameNet, se mejora VerbNet con la semántica de los marcos y se implementan las restricciones de selección haciendo uso de las clases semánticas existentes en WordNet.

5. Importando Recursos Cercanos

Cuando queremos afrontar la tarea de crear un recurso lingüístico, una posibilidad que tenemos al alcance de nuestra mano en muchos casos, es adaptar otro recurso "cercano" al que deseamos crear. Es la opción elegida por ejemplo en (Carreras, 2003), donde se construye un reconocedor de entidades con nombre para el catalán partiendo de recursos en castellano. Se emplean dos vías para lograrlo: en primer lugar creando los modelos para el español para posteriormente traducirlos al catalán, y en segundo lugar crear los modelos de forma bilingüe directamente.

La cercanía en este caso se presenta ya que se trata de dos lenguas románicas que poseen estructuras sintácticas similares y cuyos entornos sociales y culturales se solapan en gran medida, haciendo que exista un gran número de entidades que aparecen en los corpus de ambas lenguas. Estas características hacen que los recursos en español sean aprovechables para llevar a cabo tareas sobre el catalán como puede ser el reconocimiento de entidades con nombre.

Para el estudio que se llevó a cabo en este caso, se asumen dos puntos: las entidades aparecen en los mismos contextos para ambas lenguas y las entidades responden a los mismos patrones en ambos casos. Además de esto se construye un diccionario sencillo de palabra a palabra sin tener en cuenta el contexto (10 horas de trabajo para la version catalan-español y un sistema automático para la versión español-catalán).

Teniendo en cuenta estas premisas se llevan a cabo varios experimentos sobre el reconocimiento de entidades con nombre en catalán partiendo de corpus etiquetados únicamente en español.

La primera opción es traducir el modelo que se genera al entrenar con los textos en español, de manera que se analizan los árboles de decisión generados para su posterior modificación. Si un nodo del árbol analiza la posibilidad de que en la posición -2 aparezca la palabra "calle", se traduce dicho nodo haciendo lo mismo para la palabra "carrer" (traducción del español al catalán). De esta forma se puede aplicar un modelo creado mediante corpus en español a un texto en catalán. La traducción se hará en todos los nodos que analicen características léxicas del texto, mientras que los demás permanecerán intactos.

Una segunda opción es utilizar características bilingües (denominadas cross-linguistic features) basadas en una entrada del diccionario "es_w $\sim$ ca_w" (suponiendo que existe un parámetro 'lang' de valor 'es' para el español y 'ca' para el catalán). Estas características binarias se comportan de la siguiente forma:

$$\text{X-Ling}_{\text{es\_w}\sim\text{ca\_w}}\ (w) =$$

$$= \begin{cases} 1 & \text{if } w = \text{es\_w and } lang = \text{es} \\ 1 & \text{if } w = \text{ca\_w and } lang = \text{ca} \\ 0 & \text{otherwise} \end{cases}$$

De esta forma se puede entrenar el modelo con ejemplos mezclados en ambos idiomas, pudiendo seleccionar el número de ejemplos de cada caso y permitiendo por ejemplo que haya un número muy reducido de ejemplos en catalán para este escenario en concreto. El resultado es un modelo que puede reconocer entidades tanto en español como en catalán.

La tercera opción consiste por último en crear el modelo entrenando con un pequeño corpus del idioma para el que se desea ejecutar el reconocedor, en este caso, el catalán. En este trabajo se hizo empleando el mismo esfuerzo que se realizó para crear el diccionario, es decir, unas 10 horas de trabajo, obteniendo un pequeño corpus etiquetado.

Los resultados aportados (Carreras, 2003) demuestran que la tercera opción es la que peor responde ya que es preferible traducir los modelos o crearlos de forma que sean bilingües, antes que aprender de un número tan reducido de ejemplos. En cuanto a las otras dos opciones, la segunda se revela como la más interesante ya que, aunque sobre el español se obtienen mejores resultados con el modelo entrenado únicamente con ejemplos en español, la opción de crear un modelo bilingüe no está muy lejos en cuanto a números en español y supera de forma considerable a los demás en catalán.

Estos experimentos demuestran que se pueden aprovechar recursos "cercanos" a los que necesitamos para llevar a cabo tareas obteniendo buenos resultados con un coste bastante reducido (sobre todo en comparación al que habría que afrontar creando nuevos recursos desde cero).

Concretamente las conclusiones aportadas por los autores de este trabajo son las siguientes:

- Es mejor traducir un modelo entrenado en español que crear un pequeño corpus anotado con el que entrenar el modelo directamente en catalán.
- La traducción se puede llevar a cabo de forma automática sin pérdida considerable de efectividad en el proceso.
- La mejor opción ha resultado ser el uso de características bilingües ya que permite obtener resultados favorables en ambos idiomas.

La expansión de esta idea puede venir en forma de aplicaciones de apoyo más complejas y que ayuden a acercar recursos que no estén tan estrechamente ligados como los que aquí se han comentado.

6. *Técnicas de Bootstrapping*

En otros trabajos se pone en práctica otra técnica de obtención de recursos muy interesante. Se trata de las técnicas de bootstrapping, que tratan de obtener una gran cantidad de material partiendo de una pequeña

"semilla". En la tarea de la creación de corpus etiquetados, el objetivo será obtener un gran número de frases etiquetadas de forma automática partiendo de un número muy reducido de frases etiquetadas manualmente (por lo que el coste es muy bajo en comparación con el etiquetado manual completo).

Existen múltiples técnicas de bootstrapping, que difieren en la forma de aumentar la semilla, el manejo de las frases nuevas etiquetadas o las técnicas de selección en caso de utilizarse alguna. En cualquier caso todas responden a la definición:

> "la elevación de un pequeño esfuerzo inicial hacia algo más grande y más significativo".

Algunos de los esquemas de ejecución más populares dentro de las conocidas como técnicas de bootstrapping son:

- Self-train: Un corpus es utilizado para crear un modelo que se aplica a un conjunto nuevo de frases que tras ser etiquetadas pasan a formar parte del corpus original para volver a generar un nuevo modelo y avanzar de esta forma iterativamente.

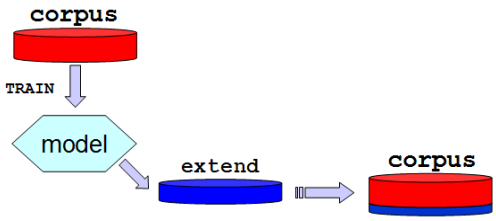

Figura 2: Esquema de ejecución para el 'self-train'.

  Esta es la definición de self-training que generalmente se adopta, como en (Clark, 2003), aunque existen otras como la que aporta (Ng, 2003), donde se describe como el entrenamiento de un comité de clasificadores utilizando bagging para finalmente utilizar la votación por mayoría para seleccionar las etiquetas finales.

- Collaborative-train: Se emplea un mismo corpus para obtener diferentes modelos empleando diferentes técnicas de aprendizaje. Posteriormente se introduce una fase de selección entre las diferentes opiniones que surgen de aplicar estos modelos al conjunto de frases nuevas y las etiquetas seleccionadas sirven para aumentar el corpus original y proseguir con la siguiente iteración.

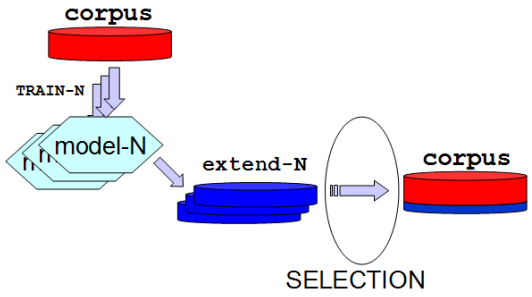

Figura 3: Esquema de ejecución para el 'collaborative-train'.

- Co-train: Dos corpus inicialmente iguales sirven para crear dos modelos de diferentes características y los resultados de aplicar estos modelos a un conjunto de frases nuevas se "cruzan", es decir, las frases etiquetadas por el primer modelo sirven para aumentar el corpus que sirvió para crear el segundo modelo y viceversa. De esta forma un modelo no se alimenta únicamente de su percepción del corpus sino que recibe información de otro modelo que imprime otro punto de vista diferente a la resolución del mismo problema.

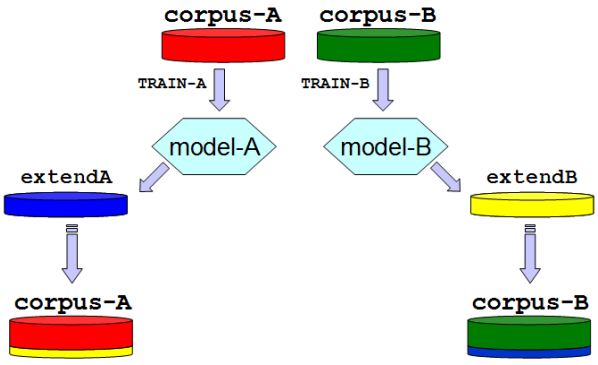

Figura 4: Esquema de ejecución para el 'co-train'.

En (Jones, 1999) se presentan dos casos de estudio para el uso de técnicas de bootstrapping en la creación de recursos. Se trata de un reconocedor de localizaciones y un clasificador de artículos de investigación. En ambos casos se obtienen muy buenos resultados, mostrando la utilidad de este tipo de técnicas.

Otro aspecto importante a tener en cuenta es que se hace prácticamente imposible mejorar el resultado de un clasificador si los resultados que alcanza son demasiado buenos. En estos casos la aplicación de estas técnicas se limitará a introducir ruido y empeorar la calidad del trabajo resultante. Es por lo tanto necesario reservar este tipo de técnicas a trabajos "difíciles" como puede ser aumentar un corpus que solo contiene un número limitado de frases inicialmente, teniendo en cuenta que si el tamaño inicial es suficiente para obtener buenos resultados, difícilmente podremos mejorarlos aplicando bootstrapping.

7. Conclusiones

La disponibilidad de recursos es un factor crucial en muchas de las tareas del Procesamiento del Lenguaje Natural que se resuelven fundamentalmente mediante métodos de aprendizaje supervisado. La obtención de estos recursos es una labor muy costosa, de ahí que se lleven a cabo esfuerzos para desarrollar métodos que desempeñen esta labor de forma automática o semi-automática. Hemos presentado varias iniciativas ya existentes, mostrando las características propias de cada una de ellas y reflejando diferentes enfoques que creemos pueden llegar a compaginarse en un entorno que facilite la tarea de la generación de recursos. Este es el punto de partida de una linea de trabajo futuro que deseamos recorrer y de la que intentaremos extraer soluciones satisfactorias a este problema.

Bibliografía

H.T. Ng: Getting serious about word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*. (1997) 1–7

R. Mihalcea: Bootstrapping Large Sense Tagged Corpora. In *Proceedings of the 3rd International Conference on Languages Resources and Evaluations*. (2002)

G. Miller, C. Leacock, T. Randee, R. Bunker: A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*. (1993) 303–308

G. Miller: Wordnet: A lexical database. Communication of the ACM,38(11). (1995) 39–41

J. Howe: The rise of crowdsourcing. Wired - 14.06 http://www.wired.com /wired/archive/14.06/crowds.html. (2006) 17–20

D. Stork: The Open Mind initiative. IEEE Expert Systems and Their Applications, 14(3). (1999) 19–20

R. Mihalcea, T. Chklovski: Open Mind Word Expert: Creating Large Annotated Data Collections with Web Users' Help. In *Proceedings of the EACL 2003 Workshop on Linguistically Annotated Corpora (LINC 2003)*. (2003) 17–20

P. Singh, T. Lin, E. Mueller, G. Lim, T. Perkins, W. Li Zhu: Open mind common sense: Knowledge acquisition from the general public. In *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems*. (2002)

Lei Shi, Rada Mihalcea: Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*. (2005)

Xavier Carreras, Lluís Màrquez, Lluís Padró: Named Entity Recognition for Catalan Using Spanish Resources. In *10th Conference of the European Chapter of the Association for Computational Linguistics*. (2003)

S. Clark, J. R. Curran, M. Osborne: Bootstrapping POS taggers using Unlabelled Data. In *Proceedings of CoNLL-2003*. (2003) 49–55

V. Ng, C. Cardie: Weakly supervised natural language learning without redundant views. In *Human Language Technology/Conference of the North American Chapter of the Association for Computational Linguistics*. (2003)

Rosie Jones, Andrew McCallum, Kamal Nigam, Ellen Riloff: Bootstrapping for Text Learning Tasks. In *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*. (1999)