Human Query Language

Claudio Zamoszczyk •, Sebastián De Luca, Sebastián Ruiz Martínez, Lucas Iturbide ••

Resumen

El presente trabajo describirá el desarrollo de una herramienta que aprovecha el procesamiento del lenguaje natural, a efectos de alcanzar una mejora en la comunicación usuario - máquina. El proyecto, que cuenta con una interfaz web y para smartphones (teléfonos inteligentes), será capaz de interpretar consultas ingresadas por teclado o por voz, y a partir de dichas entradas generar consultas SQL capaces de ser procesadas en un motor de base de datos. El resultado de la búsqueda será presentada en forma de texto, lista o gráfico permitiendo la integración con cualquier software de gestión de negocios, típicamente ERP o CRM, pudiendo extenderse su uso a otras áreas de actividad.

Palabras claves: procesamiento del lenguaje, pln, sql, android, reconocimiento de voz.

Abstract

This paper describes the development of a tool that take advantage of the natural language processing, in order to achieve an improvement during the machine-human communication. This project, which has a web and mobile (smartphones) interface, will be able to interpret queries entered by keyboard or voice. From those entries, it will generate SQL queries capable of being processed in a database engine. The result of that query will be displayed in text format, list or graphic, allowing the integration with any business management software (ERP or CRM typically). Being able to extend its use to other areas of activity.

Keywords: language processing, npl, sql, android, voice recognition

Fecha de Recepción: octubre 2012 | Fecha de Aceptación: noviembre 2012

• Lic. en Informática. Docente e investigador – Universidad de Palermo

•• Alumnos de la Facultad de Ingeniería UP

I. Introducción

Procesamiento de lenguajes naturales. (PLN)

El procesamiento del Lenguaje Natural es una disciplina que relaciona directamente la informática con la lingüística. La misma persigue como objetivo, poder conseguir que el lenguaje coloquial (el lenguaje de uso cotidiano de todos nosotros) pueda ser utilizado como una entrada en un sistema informático [1]. Es importante poder destacar entre tres tipos de objetivos que persigue el procesamiento de lenguaje natural:

1. Interfaces en lenguaje natural: Lograr la comunicación con distintos dispositivos a través del lenguaje natural.

2. Procesamiento de textos: Se refiere a lograr extraer datos significativos de textos escritos en lenguaje natural, a efectos de realizar el procesamiento de los mismos. Esto intenta abordar un inconveniente del mundo actual, en donde la mayor cantidad de información se encuentra almacenada en forma de texto. Esto implica que la información de los mismos no puede ser procesada de forma directa. Ejemplos de esto puede ser bases de datos relacionales o registros de transacciones bancarias.

3. Traducción automática: Es el objetivo original del PLN, que consta del análisis y tratamiento de lenguaje natural por medio de la utilización de herramientas tanto lingüísticas como informáticas.

Historia de los sistemas PLN

El PLN es una de las piedras angulares tempranas de la Inteligencia Artificial (IA). La Traducción automática, por ejemplo, nació a finales de la década de los cuarenta, antes de que se acuñara la propia expresión «Inteligencia Artificial». No obstante, el PLN ha desempeñado múltiples papeles en el contexto de la IA, y su importancia dentro de este campo ha crecido y decrecido a consecuencia de cambios tecnológicos y científicos [2].

Entre la década del cuarenta y cincuenta, se realizaron los primeros intentos de traducción de textos, los cuales fracasaron debido a la escasa potencia de las computadoras y a la escasa sofisticación lingüística. Sin embargo, en la década de los sesenta se empezaron a obtener un cierto grado significativo de éxito, en la construcción de interfaces basadas en lenguaje natural, para diversas aplicaciones

informáticas. En la década de los ochenta y principios de los noventa resurge la investigación dentro del área de la traducción automática. Este progreso favorable se debe a una combinación de factores que van desde un enorme aumento en la potencia de los procesadores en relación a su coste hasta modelos del lenguaje humano mejores y más susceptibles de ser tratados computacionalmente. Por otra parte, nunca ha sido mayor la necesidad de sistemas PLN para procesar datos textuales, incluyendo traducción, clasificación, recuperación y extracción de información.

Arquitectura de un sistema PLN

Es muy importante el análisis de la arquitectura de un sistema PLN para comprender su funcionamiento. En la misma se expresa cómo interactúan el usuario con la máquina y los pasos internos realizados para el análisis de la información. A continuación se enumerarán los pasos en la tarea de procesamiento del lenguaje [2]:

1. El usuario le expresa (de alguna forma) a la computadora el texto que desea procesar.
2. La computadora analiza las oraciones proporcionadas, en el sentido morfológico y sintáctico.
3. Luego, se analizan las oraciones semánticamente, es decir se determina el significado de cada oración.
4. Por ultimo se realiza el análisis pragmático del texto. De esta forma se obtiene una expresión final que luego es utilizada directamente con un fin determinado.

Si bien se observa que son un conjunto de pasos reducidos, la complejidad radica en el análisis de las palabras y su contexto. Por dicha cuestión es necesario realizar un conjunto de análisis más excautivos con el fin de comprender la oración en su totalidad. Dichos análisis se pueden enumerar de la siguiente forma [3]:

- *Análisis morfológico*: El análisis de las palabras para extraer raíces, rasgos flexivos, unidades léxicas compuestas y otros fenómenos. Clasificar las palabras según la categoría gramatical. Selección de atributos relevantes para la consulta. Analizar variantes y posibles alternativas para cada atributo. Realizar la mayor cantidad de combinaciones posibles para determinar cuál es la más apropiada. En todos los casos se considera el orden en el que las palabras aparecen en la frase.

- *Análisis semántico*: La extracción del significado de la frase, y la resolución de ambigüedades léxicas y estructurales. Determinar el significado de cada palabra dentro de la oración.
- *Análisis pragmático*: El análisis del texto más allá de los límites de la frase, por ejemplo, para determinar los antecedentes referenciales de los pronombres. Identificar el tipo de instrucción que ingresó el usuario.

Dificultades en los sistemas PLN

El lenguaje natural, posee propiedades que afectan la efectividad de los sistemas PLN. Estas propiedades son la variación y la ambigüedad lingüística. Cuando hablamos de la variación lingüística nos referimos a la posibilidad de utilizar diferentes palabras o expresiones para comunicar una misma idea. En cambio, la ambigüedad lingüística se produce cuando una palabra o frase permite más de una interpretación [4].

Ambos fenómenos inciden de forma distinta en el proceso de recuperación de la información. La variación lingüística provoca el silencio documental, es decir la omisión de documentos relevantes para cubrir la necesidad de información, ya que no se han utilizado los mismos términos que aparecen en el documento. En cambio, la ambigüedad implica el ruido documental, es decir la inclusión de documentos que no son significativos, ya que se recuperan también documentos que utilizan el término pero con significado diferente al requerido. Estas dos características dificultan considerablemente el tratamiento automatizado del lenguaje. A continuación se muestran unos ejemplos sobre los diversos casos que podemos encontrar:

Ejemplo 1. "Deja la comida que sobre sobre la mesa de la cocina, dijo llevando el sobre en la mano."

La palabra "sobre" es ambigua morfológicamente ya que puede ser un sustantivo masculino singular, una preposición, y también la primera o tercera persona del presente de subjuntivo del verbo sobrar.

A nivel sintáctico, centrado en el estudio de las relaciones establecidas entre las palabras para formar unidades superiores, sintagmas y frases, se produce ambigüedad a consecuencia de la posibilidad de asociar a una frase más de una estructura sintáctica.

Ejemplo 2. "María vio a un niño con un telescopio en la ventana."

La interpretación de la dependencia de los dos sintagmas preposicionales (conjunto de palabras que forman una unidad dentro de una oración), con un telescopio y en la ventana, otorga diferentes significados a la frase: (i) María vio a un niño que estaba en la ventana y que tenía un telescopio, (ii) María estaba en la ventana, desde donde vio a un niño que tenía un telescopio, y (iii) María estaba en la ventana, desde donde miraba con un telescopio, y vio a un niño.

Ejemplo 3. "Luís dejó el periódico en el banco."

El término banco puede tener dos significados en esta frase, (i) entidad bancaria y (ii) silla. La interpretación de esa frase va más allá del análisis de los componentes que forman la frase, se realiza a partir del contexto en que es formulada.

Ejemplo 4. "Ella le dijo que los pusiera debajo"

Otro factor importante es la ambigüedad provocada por la presencia en la oración de pronombres y adverbios que hacen referencia a algo mencionado con anterioridad. La interpretación de esta frase tiene diferentes incógnitas ocasionadas por la utilización de pronombres y adverbio: ¿quién habló?, ¿a quién?, ¿qué pusiera qué?, ¿debajo de dónde?. Por tanto, para otorgar un significado a esta frase debe recurrirse nuevamente al contexto en que es formulada.

Debido a estos ejemplos, y muchos otros que pudiéramos mencionar, queda claro que la tarea de procesar lenguaje normal de forma automática, no es para nada sencilla.

II. Proyecto Human Query Language

Introducción y características

El proyecto que trata este documento consiste en la integración de diversas herramientas informáticas y lingüísticas, con el fin de lograr un sistema que acepte consultas en lenguaje natural, ya sean ingresadas a través de teclado o por voz, y a partir de dichas consultas poder generar y ejecutar un SQL capaz de ser procesado por cualquier base de datos.

Posibles aplicaciones y usos:

- Motor de consultas natural para software de terceros. Ej: Bases de datos, CRMs, ERPs, entre otros.

- Sistemas autónomos de atención al público activados por voz o por texto. Ej: Kioscos de atención, Call centers.
- Puntos de venta virtuales.
- Herramienta de accesibilidad para personas con algun grado de discapacidad.
- Sistemas de sugerencia de corrección de errores y auto completamiento de palabras.

El sistema y su funcionamiento

El sistema cuenta con diversas interfaces graficas de fácil interacción con el usuario, las cuales permiten ingresar consultas, como mostrar las respuestas de las mismas ya sea mediante texto, gráficos o tablas, de acuerdo a como sea solicitado. Estas interfaces se encuentran desarrolladas en HTML 5 en combinación con diversas bibliotecas de JQuery (framework javascript) y en Google Android. (Figuras 1, 2 ,3 y 4)

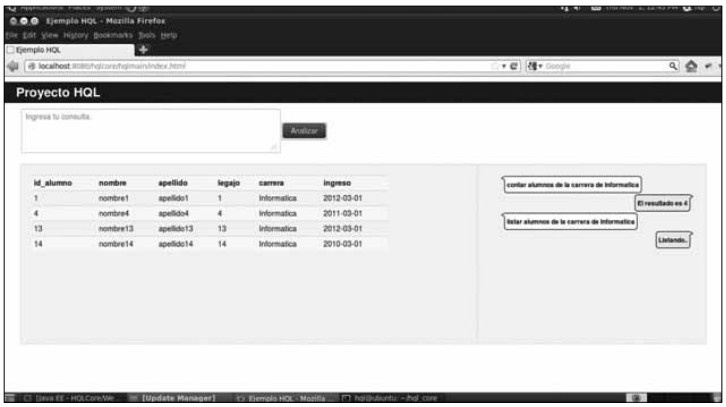

Figura 1: Pantalla inicial de la aplicación web para realizar consultas mediante texto

En el caso del cliente web, se utiliza una interfaz simple donde el usuario ingresa el texto en un formulario. A medida que escribe, éste le sugiere palabras que el sistema puede interpretar. Este procedimiento es posible a partir de la utilización del lenguaje de programación "javascript", combinado con las mencionadas bibliotecas y técnicas de programación asíncronas (AJAX), que no interrumpen al usuario a medida que escribe. El resultado de la consulta es mostrado en formato de texto, tablas o gráficos.

En el caso de la interfaz para smartphone, la misma se encuentra desarrollada en Android en su totalidad, utilizando el servidor HQL (Servidor central de

procesamiento) como nexo entre el sistema PLN y el usuario. El sistema Android utiliza el propio sistema operativo para generar texto a partir de la voz STT (Speech-to-Text), de esta manera se puede apretar un botón y proceder al análisis de la consulta. También tiene la opción de escribir el texto. Los resultados que ofrece el servidor son mostrados al usuario en forma de texto (verbalmente usando TTS text-to-speech), gráfico o tablas. (Figuras 2 ,3 y 4)

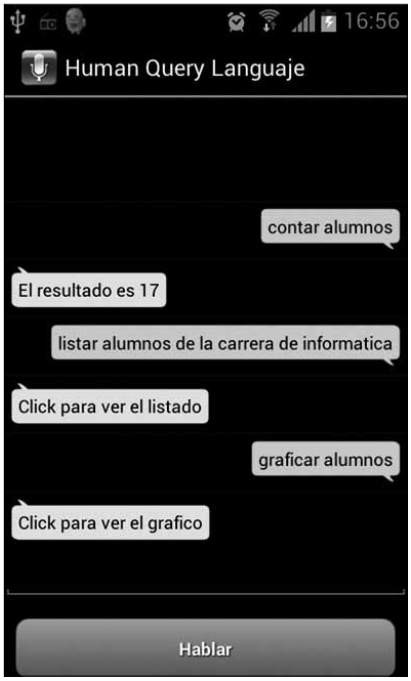

Figura 2: Pantalla inicial de la aplicación mobile para realizar consultas mediante voz.

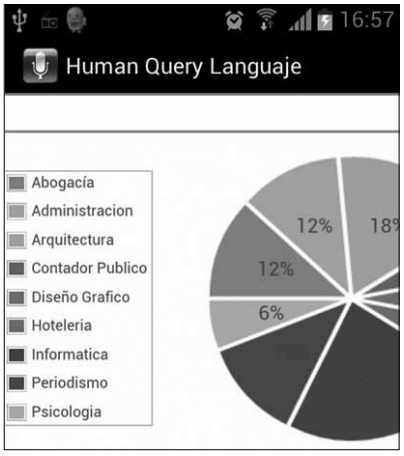

Figura 3: Ejemplo de gráfico

Figura 4: Ejemplo de listado

En ambos casos, las consultas son enviadas al servidor HQL, por medio de solicitudes HTTP, utilizando el Api REST (Representational state transfer), en donde serán pre-procesadas. Este pre-procesamiento implica encontrar palabras claves, reemplazar términos y demás cuestiones semánticas del lenguaje natural. Una vez realizado el procesamiento se comienza a trabajar en los términos encontrados para determinar con qué tablas, columnas y condiciones se tiene que trabajar, y finalmente generar la sentencia SQL.

Luego, con esta consulta que generamos, se procede a ejecutar la misma en la base de datos. Los datos obtenidos, serán transmitidos a la interfaz en donde se realizó la consulta inicialmente, para así poder ser visualizados por el usuario.

Componentes

A continuación procederemos a explicar algunos componentes importantes del sistema.

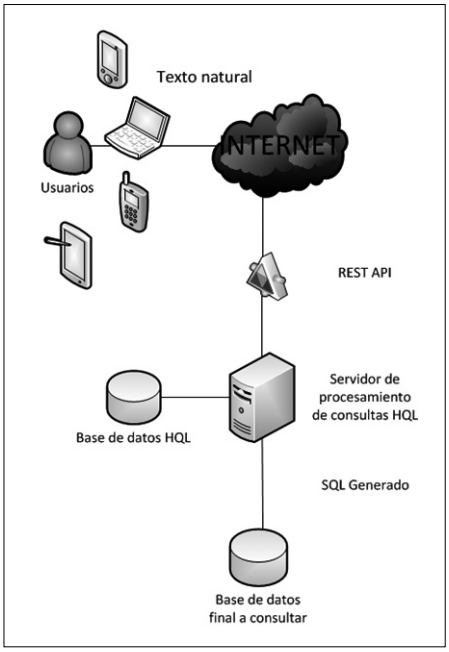

Figura 5: Esquema general del sistema

Freeling

FreeLing es una librería de código abierto para el procesamiento multilingüe, que proporciona una amplia gama de funcionalidades de análisis para varios idiomas. El proyecto FreeLing, iniciado desde el centro TALP (Centro de Tecnologías y Aplicaciones del Lenguaje y del Habla) de la Universitat Politècnica de Catalunya, tiene como objetivo avanzar hacia la disponibilidad general de recursos y herramientas básicos de PLN [6]. Esta disponibilidad busca posibilitar avances más rápidos en proyectos de investigación y desarrollo de PLN.

FreeLing, debido a que se encuentra estructurado como una biblioteca, permite ser llamado desde cualquier aplicación de usuario que requiera servicios de análisis de lenguaje.

Se eligió utilizar dicha biblioteca como soporte para el análisis de lenguaje natural, ya que por sus características y arquitectura, permite que sea una herramienta muy veloz al momento de procesar información lingüística. Otro aspecto importante es que permite integrarse con Java. Además, es compatible con el idioma español, entre otros. Otra ventaja muy importante de esta biblioteca es que nos brinda una lista muy variada de servicios de análisis disponibles para los diversos idiomas que soporta (hay que destacar

que algunos servicios no se encuentran disponibles para algunos idiomas). Algunos de los servicios son: separación de la oración, detección de fechas y números, detección de palabras múltiples, detección básica de entidades, clasificación de entidades, entre muchos otros servicios que son de utilidad. Un ejemplo de como funciona Freeling puede ser encontrado en la siguiente dirección http://nlp.lsi.upc.edu/freeling/demo/demo.php.

Apache Lucene

Apache Lucene es una novedosa herramienta que permite tanto la indexación como búsqueda de texto libre sobre documentos. Escrita en Java y distribuida mediante un Api permite realizar diferentes tipos de búsquedas semánticas. Una de sus características más importantes es la búsqueda difusa.

La búsqueda difusa sirve para encontrar correspondencias aunque las palabras usadas tengan errores de ortografía, no estén completas o se acercan a un tipo de palabra esperada. Este tipo de búsqueda es fundamental para el procesamiento de texto, ya que somos propensos a cometer errores.

Api REST

REST (Representational state transfer) define un set de principios arquitectónicos por los cuales se diseñan servicios web haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacía clientes escritos en diversos lenguajes de programación. REST emergió en los últimos años como el modelo predominante para el diseño de servicios. De hecho, REST logró un impacto tan grande en la web que prácticamente logró desplazar a SOAP (Simple Object Access Protocol) y las interfaces basadas en WSDL (Web Services Description Language) por tener un estilo bastante más simple de usar.

¿Qué ventajas traer utilizar una API REST para el sistema? La principal ventaja que tiene acceder al sistema a través de una API REST es la simplicidad. En general, los servicios web a los que se puede acceder a través de una interfaz REST son muy fáciles de consumir, lo cual simplifica la programación y el mantenimiento. Además, el acceso REST aumenta el desacoplamiento entre el sistema y los clientes que consumen estos servicios, salvaguardando a los clientes de la posible evolución de los mismos.

HQL Server

La responsabilidad del HQL Server es la de encapsular la lógica del negocio del sistema interactuando con los diferentes componentes. En líneas generales actúa como una fachada entre el sistema y el exterior del mismo.

Figura 2.5: Diagrama de componentes del sistema

Integración de los componentes del sistema

Tradicionalmente cada objeto es responsable de obtener sus propias referencias a los objetos con los que colabora. Este modelo de trabajo trae aparejado una problemática de acoplamiento entre los elementos. Para solucionar esto existe el concepto de Inyección de dependencias.

La Inyección de Dependencia (en inglés Dependency Injection, DI) es un patrón de diseño orientado a objetos, en el que se inyectan objetos a una clase en lugar de ser la propia clase quien cree el objeto.

La forma habitual de implementar este patrón es mediante un "Contenedor DI". El contenedor inyecta a cada objeto los objetos necesarios según las relaciones plasmadas en un archivo de configuración. Típicamente este contenedor es implementado por un framework externo. Para el sistema propuesto se utilizo Spring.

Herramientas

Como ya mencionamos anteriormente, se utilizaron diversas herramientas para lograr el desarrollo del Sistema. A Continuación brindamos una lista de las herramientas utilizadas:

- Java 1.7
- Hibernate 3.4
- Hibernate Search 3.0 con Apache Lucene
- Eclipse EE Ide
- Spring MVC 3.0
- Spring Core 3.0
- Freeling 3.0
- HTML5
- JQuery
- Twitter Bootstrap
- Apache Tomcat 7.0
- Apache Lucene
- Ubuntu Linux 10
- Api REST
- Base de datos Mysql 5.5
- HQL server
- Google Voice Recognizer

Ejemplo de consultas

A continuación se enumerarán posibles consultas que el sistema soporta, junto con el SQL que se genera.

"Listar alumnos"
SELECT id_alumno, nombre, apellido, legajo, carrera, fecha_ingreso, promedio FROM alumnos

"Listar alumnos ordenados por promedio"
SELECT id_alumno, nombre, apellido, legajo, carrera, fecha_ingreso, promedio FROM alumnos ORDER BY promedio

"Listar alumnos ordenados por promedio, carrera y apellido"
SELECT id_alumno, nombre, apellido, legajo, carrera, fecha_ingreso, promedio FROM alumnos ORDER BY promedio, carrera, apellido

"Listar alumnos de la carrera de informática"
SELECT id_alumno, nombre, apellido, legajo, carrera, fecha_ingreso, promedio FROM alumnos WHERE carrera = 'informatica'

"Listar alumnos con fecha_ingreso entre marzo de 2009 y agosto de 2010"
SELECT id_alumno, nombre, apellido, legajo, carrera, fecha_ingreso, promedio FROM alumnos WHERE fecha_ingreso >= '2009-03-01' AND fecha_ingreso <= '2010-08-01'

"Graficar alumnos"
SELECT carrera, COUNT(carrera) as value FROM alumnos GROUP BY carrera

"Graficar alumnos de la carrera de arquitectura y hotelería"
SELECT carrera, COUNT(carrera) as value FROM alumnos WHERE (carrera = 'arquitectura' OR carrera = 'hoteleria') GROUP BY carrera

"Listar alumnos donde carrera es igual a informática y promedio es igual a 4"
SELECT id_alumno, nombre, apellido, legajo, carrera, fecha_ingreso, promedio FROM alumnos WHERE carrera = 'informatica' AND promedio = 4

"*Listar alumnos de la carrera de informática y con promedio entre 7 y 10"*
SELECT id_alumno, nombre, apellido, legajo, carrera, fecha_ingreso, promedio FROM alumnos WHERE carrera = 'informatica' AND promedio >= 7 AND promedio <= 10

"Listar alumnos donde legajo es mayor a 70000"
SELECT id_alumno, nombre, apellido, legajo, carrera, fecha_ingreso, promedio FROM alumnos WHERE legajo > 70000

III. Conclusión

El procesamiento del lenguaje natural tiene como objetivo fundamental lograr una comunicación maquina - humano similar a la comunicación humano-humano.

El empleo del lenguaje le permite al hombre trasmitir sus conocimientos, sentimientos, sensaciones, emociones, y estados de ánimo. A lo largo de la historia los lenguajes naturales han ido evolucionando, de forma paralela al desarrollo y evolución de la especie humana.

El proyecto HQL intenta demostrar que es factible desarrollar una solución simple que permita interactuar mediante en lenguaje natural con otros sistemas, alcanzando los mismos resultados que por medio de lenguajes formales.

IV. Líneas futuras de investigación

- Creación de una gramática para mejorar tanto el análisis como el tratamiento de las consultas ingresadas.
- Extensión del proyecto HQL para trabajar con otros idiomas de entrada. Ej: Ingles

V. Agradecimientos

Al Ing. Esteban Di Tada y a la Lic. Adriana Álvarez por su continuo apoyo en el proceso de investigación y desarrollo de la solución propuesta.

VI. Referencias

[1] Mario Alberich, "Procesamiento del Lenguaje Natural - Guía Introductoria". Disponible en http://www.sopadebits.com/wp-content/uploads/2011/03/4479-pln-1.0-20070630.pdf.

[2] Jaime Carbonell, Carnegie Mellon University, "El procesamiento del lenguaje natural, tecnología en transición". Disponible en http://cvc.cervantes.es/obref/congresos/sevilla/tecnologias/ponenc_carbonell.htm.

[3] Ernesto González Díaz, "Procesamiento del lenguaje natural en la Inteligencia Artificial". Disponible en http://www.monografias.com/trabajos17/lenguaje-natural/lenguaje-natural.shtml

[4] Mari Vallez (Universitat Pompeu Fabra) y Rafael Pedraza-Jimenez (Universitat Pompeu Fabra), "El Procesamiento del Lenguaje Natural en la Recuperación de Información Textual y áreas afines". Disponible en http://www.upf.edu/hipertextnet/numero-5/pln.html

[5] Wikipwdia, "Procesamiento de lenguajes naturales". Disponible en http://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales#Ambig.C3.BCedad

[6] Lluís Padró, "Analizadores Multilingues en FreeLing". Disponible en http://nlp.lsi.upc.edu/publications/papers/padro11.pdf.