Síntesis de Trabajo Final de Grado

Revista Difusiones, ISSN 2314-1662, Num. 12, Agosto 2017, p96-118

Etiquetado automático mediante Programación de Lenguaje Natural (PNL) y visualización de la información utilizando repositorios web bajo protocolo OAI-PMH

Autor: Fernando R. Aramayo
(1) Universidad Nacional de Jujuy, Facultad de Ingeniería
(2) Universidad Nacional de Jujuy, Facultad de Ciencias Agrarias
fernando.ruben.aramayo@gmail.com

Ingeniero en Informática (Universidad Nacional de Jujuy), Licenciado en Sistemas (Universidad Nacional de Jujuy), Profesor Adjunto en la Facultad de Ciencias Agrarias de la UNJU.

Resumen

El presente trabajo propone el desarrollo de un sistema que logre procesar información que se encuentra en repositorios de la Web los cuales son de libre acceso, comúnmente a estos tipos de sistemas se los denomina “Harvester”. Son sistemas clientes que manejan llamadas y respuestas a los repositorios, los cuales se encuentran en algunas de estas plataformas Archimede, ARNO, CDSware, DSpace, Fedora, Eprints, i-Tor, Mycore y OPUS, etc. las cuales son las encargadas de proveer datos y servicios. Las mismas utilizan el protocolo OAI-PMH para poder establecer la comunicación con los sistemas que extraen

información de ellas, su arquitectura está basada en clientes y servidores. Los primeros son los archivos que proporcionan la información, y los segundos son los recolectores o servicios que toman los datos, con el objetivo de incorporarles algún valor añadido y presentarlos a los usuarios finales.

Se analizaron los repositorios listados en ROAR (listado de repositorios) y se observó que, si bien cada repositorio tiene su propio sistema de búsqueda, la misma se establece analizando el nombre de los documentos que los usuarios hayan subido a dicha plataforma y en algunas ocasiones con solo esa información no es suficiente para poder realizar una búsqueda óptima.

El sistema debe obtener la información de cada uno de los recursos que se encuentran en un determinado repositorio, interpretar/procesar la información obtenida y etiquetar ese recurso automáticamente, la o las etiquetas resultantes de este procesamiento permitirán el agrupado de los recursos formando conjuntos o clúster con contenidos similares.

Palabras clave

Repositorios, protocolo OAI-PMH, harvester, programación Lenguaje Natural (PNL).

Automatic tagging using Natural Language Programming (NLP) and visualization of information using web repositories under OAI-PMH protocol

Abstract

The present work proposes the development of a system that can process information that is found in repositories of the Web which are of free access, commonly to these types of systems are called "Harvester". They are client systems that handle calls and replies to the

repositories, which are found in some of these platforms Archimede, ARNO, CDSware, DSpace, Fedora, Eprints, i-Tor, Mycore and OPUS, etc. which are in charge of providing data and services. They use the OAI-PMH protocol to establish communication with systems that extract information from them, its architecture is based on clients and servers. The first are the files that provide the information, and the second are the collectors or services that take the data, in order to incorporate some added value and present them to the users.

We analyzed the repositories listed in ROAR (list of repositories) and we observed that although each repository has its own search system, it is established by analyzing the name of the documents that the users have uploaded to that platform and in some occasions with only that information is not sufficient to perform an optimal search.

The system must obtain the information of each of the resources that are in a given repository, interpret / process the obtained information and label that resource automatically, the resulting labels or tags will allow the grouping of the resources forming groups or clusters with similar contents.

Key Words

Repositories, OAI-PMH protocol, harvester, Natural Language programming (NLP).

Introducción

El problema que se analiza pertenece a una rama de la Recuperación de la Información denominada "Etiquetado Automático" (Cañada, 2006), el cual se presenta como un modelo para organizar, describir y compartir recursos web, también denominado tagging.

Algunos autores asocian el tagging o los tags a categorías mientras que otros a palabras clave. La distinción conceptual entre palabras clave y categorías, y por tanto entre descripción y clasificación, es básicamente una cuestión de especificidad. Las categorías representan la temática global bajo la que se enmarca un recurso, mientras que las palabras clave describen aquellos conceptos que son tratados en el recurso (Hassan y Núñez-Peña, 2005; Hassan, 2006).

El presente documento propone el desarrollo de un sistema que logre procesar información que se encuentra en repositorios de la Web los cuales son de libre acceso, comúnmente a estos tipos de sistemas se los denomina “Harvester” o “Cosechador”. Los Harvester son sistemas clientes que manejan llamadas y respuestas a los repositorios, los cuales se encuentran en algunas de estas plataformas Archimede, ARNO, CDSware, DSpace, Fedora, Eprints, i-Tor, Mycore y OPUS, etc. las cuales son las encargadas de proveer datos y servicios.

Las mismas utilizan el protocolo OAI-PMH para poder establecer la comunicación con los sistemas que extraen información de ellas y su arquitectura está basada en clientes y servidores. En la Figura 1 se puede observar un esquema básico de dicho protocolo con todos los actores y elementos involucrados.

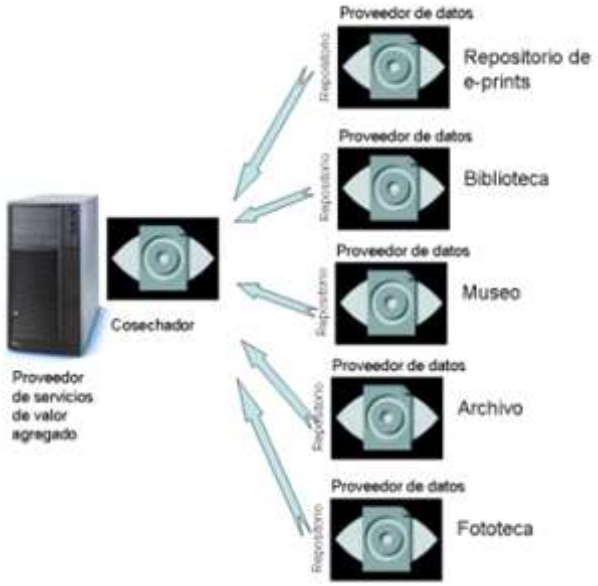

Figura 1: Esquema del protocolo OAI-PMH

La idea principal es que el sistema Harvester consuma la información proveniente de los repositorios a través del protocolo OAI-PMH, estos repositorios proveen la información de su contenido mediante un esquema XML que contiene por un lado una cabecera del recurso compuesta por 3 elementos (identificador único del registro, la fecha de último acceso al registro y opcionalmente un identificador de conjunto utilizado internamente por el repositorio) y por otro lado contiene el metadata compuesto por 15 elementos (donde los principales elementos son: el título del recurso, la descripción o resumen, la fecha y los creadores o autores). Los repositorios varían en tamaño y los recursos a los cuales hacen referencia se encuentran en distintos formatos (*.pdf, *.doc, *.jpg, etc.).

El sistema propuesto debe obtener la información de cada uno de los recursos que se encuentran en un determinado repositorio, interpretar/procesar la información obtenida y etiquetar ese recurso automáticamente (Anderson y Pérez-Carballo, 2001), la o las etiquetas resultantes de este procesamiento permitirán el agrupado de los recursos formando conjuntos o clúster con contenidos similares. Dentro de un repositorio, los recursos pueden estar separados por secciones como ser Física, Electrónica, etc., por ende, ya hay una asociación implícita en el contenido de los documentos, pero la misma es muy superficial, esta asociación implícita que se menciona no siempre se encuentra en los

repositorios, es decir, no es un requisito indispensable para implementar el protocolo OAI-PMH. Y estas secciones en la mayoría de los casos no son representativas de los documentos que contiene, siguiendo el ejemplo de los recursos que pertenecen al área de Física, en la descripción/resumen que poseen estos recursos pueden existir temas que se relacionen con otro clúster de información, es decir este recurso podría estar relacionado con información referente a un clúster de microelectrónica, química, etc. o temas mucho más específicos dentro de esa misma área como física de partículas, cinemática, electromagnetismo, etc.

El etiquetado se realizará mediante técnicas de procesamiento de lenguaje natural (PLN) (Perkins, 2010), el sistema procesará la descripción de cada uno de los registros que se obtengan del repositorio e intentará extraer de dicha descripción "descriptores significativos" que permitan luego fijar las relaciones entre los diferentes clústeres de información. El sistema almacenará en una Base de Datos, todas las relaciones entre los clústeres, obtenidos como resultado del procesamiento de los repositorios en cuestión. Como etapa final de resolución de este problema, se plantea una forma novedosa e interactiva, de presentar y representar los resultados al usuario, de manera que se puedan observar las relaciones encontradas de forma automática y descubrir nuevas mediante las representaciones visuales. La visualización planteada se realizará mediante técnicas de Visualización de Información (Ware, 2004,2008) la cual es un área dentro de la Inteligencia Artificial, definida como el proceso de pasar de representaciones gráficas a representaciones perceptivas, eligiendo las técnicas de codificación que maximicen la comprensión humana y la comunicación (Tufte, 1983,1997). El enfoque de la exploración de datos a través de la visualización busca combinar flexibilidad, creatividad y conocimiento general con grandes volúmenes de datos almacenados, a fin de facilitar la interacción directa con la información a través de la extracción de conocimientos y la realización de análisis y conclusiones (Carlis y Konstan, 1998; Card, Mackinlay y Shneiderman, 1999).

Objetivos

· **Objetivos Generales:** Desarrollar un Sistema que sea capaz de interactuar con repositorios bajo el protocolo OAI-PMH para recolectar toda la información que proveen los repositorios. Implementar técnicas de PLN para obtener etiquetas o descriptores para permitir realizar un etiquetado automático de los recursos.

· **Objetivos específicos:**

1. Analizar las distintas variantes utilizadas en cada repositorio para implementar dicho protocolo. Existe cierta libertad a la hora de implementar el mecanismo de respuesta por los repositorios, por ello se analizarán todas las posibles variantes de

manera de cubrir todo el rango de repositorios objeto de este estudio.

2. Implementar técnicas de PLN que permitan la extracción eficiente de etiquetas o descriptores que se utilizarán en el proceso de etiquetado automático. Este etiquetado permitirá la formación de clúster de recursos con contenido similar. Por otro lado, el etiquetado hará posible relacionar clústeres entre sí, formando relaciones y/o asociaciones de contenido entre clústeres.

3. Desarrollar una interfaz de búsqueda que permita ingresar la o las palabras a buscar, y extraer de la base de datos toda la información relacionada que ha sido previamente procesada en el paso anterior.

4. Plantear y desarrollar una técnica de Visualización de Información Interactiva para representar de la mejor manera los resultados obtenidos. Se deja de lado las visualizaciones y gráficos estáticos (tablas, gráfico de torta, gráfico de barra, etc.), ya que el objetivo es implementar una técnica de Visualización novedosa que permita observar las relaciones entre los resultados obtenidos.

Dublin Core

Los metadatos del formato Dublin Core tratan de incorporar a los repositorios los datos necesarios para identificar, procesar, describir y recuperar un documento a través de la Web. Cumplen un rol similar al de una ficha bibliográfica para la búsqueda de información en una biblioteca, los campos del Dublin Core o los metadatos persistidos en una base de datos, describen al objeto digital que se encuentra en algún medio de almacenamiento, lo cual permite un acceso a dicho objeto por medio de una dirección URL. La Dublin Core Metadata Initiative (DCMI) es la responsable del desarrollo, estandarización y promoción del conjunto de elementos de metadatos Dublin Core, el cual consiste de 15 definiciones (Guía de uso Dublin Core 2012). El objetivo de la DCMI es elaborar normas interoperables sobre metadatos y desarrollar vocabularios especializados en metadatos para la descripción de recursos o registros que permitan a sistemas de recuperación de información (Harvester), obtener información precisa referente a un registro particular. La DCMI pretende:

· Desarrollar estándares de metadatos para la recuperación de información en Internet a través de distintos dominios.

· Definir el marco para la interoperabilidad entre conjuntos de metadatos.

· Facilitar el desarrollo de conjuntos de metadatos específicos de una disciplina o comunidad que trabaja dentro del marco de la recuperación de información.

K odelo de Metadatos Dublin Core

```
<schematargetNamespace="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"attributeFormDefault="unqualified">

<annotation>
<documentation>
    XML Schema 2002-03-18 by Pete Johnston.
    Adjusted for usage in the OAI-PMH.
    Schema imports the Dublin Core elements from the DCMI schema for unqualified
Dublin Core.
    2002-12-19 updated to use simpledc20021212.xsd (instead of simpledc20020312.xs
</documentation>
</annotation>

<importnamespace="http://purl.org/dc/elements/1.1/"
schemaLocation="http://dublincore.org/schemas/xmls/simpledc20021212.xsd"/>

<elementname="dc"type="oai_dc:oai_dcType"/>

<complexTypename="oai_dcType">
<choiceminOccurs="0"maxOccurs="unbounded">
<elementref="dc:title"/>
<elementref="dc:creator"/>
<elementref="dc:subject"/>
<elementref="dc:description"/>
<elementref="dc:publisher"/>
<elementref="dc:contributor"/>
<elementref="dc:date"/>
<elementref="dc:type"/>
<elementref="dc:format"/>
<elementref="dc:identifier"/>
<elementref="dc:source"/>
<elementref="dc:language"/>
<elementref="dc:relation"/>
<elementref="dc:coverage"/>
<elementref="dc:rights"/>
</choice>
</complexType>

: -schema>
```

Tabla 1: Elementos Dublin Core

Los elementos Dublin Core pueden ser:
Opcionales
Las Definiciones se pueden repetir
Pueden aparecer en cualquier Orden

Como se encuentra detallado en la Tabla 1 estas definiciones tienen ciertas características por ende algunos de estos ítems no se encontrarán presentes, otros aparecerán una o más veces y no deberán respetar un orden de aparición. Las definiciones de los ítems del Dublin Core se pueden clasificar en tres secciones que indican el ámbito de la información que almacenan:

· Elementos relacionados principalmente con el contenido de un registro.

· Elementos relacionados principalmente con el registro cuando es percibido como una propiedad intelectual.

· Elementos relacionados principalmente con la instanciación del registro.

El Dublin Core puede ser considerado como un sistema de catalogación estandarizado para la clasificación de documentos digitales, con el cual se pueden realizar búsquedas rápidas y eficientes para cualquier tipo de documento digital que se encuentre en Internet.

Protocolo OAI-PMH

OAI-PMH (Open Archives Initiative - Protocol Metadata Harvesting o Iniciativa Abierta de Archivos – Protocolo de Recolección de Metadatos) es una herramienta que permite el intercambio de metadatos sobre cualquier material almacenado en soporte digital en un repositorio que soporte dicho protocolo. La OAI se creó con el fin de desarrollar y promover estándares de interoperabilidad para facilitar la difusión de contenidos en Internet y para mejorar el acceso a archivos de publicaciones electrónicas científicas, pero luego se hizo extensiva a la comunicación e intercambio de metadatos de materiales digitales.

Utiliza solicitudes HTTP (Gourley y Totty, 2002) (Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto) para que un usuario o servicio de recolección de metadatos (Harvester) pueda realizar solicitudes de información y para que un servidor o repositorio pueda proporcionarle respuestas a la solicitud realizada, dicha interoperabilidad queda representada por la Figura 2. El Harvester puede solicitar al repositorio que le envié metadatos especificando criterios de restricciones para restringir las búsquedas. Como respuesta dependiendo de la solicitud OAI-PMH realizada, el repositorio puede devolver un conjunto de registros en formato XML, el cual debe contener los ítems establecidos para el

formato Dublin Core para cada uno de los objetos descriptos en cada registro.

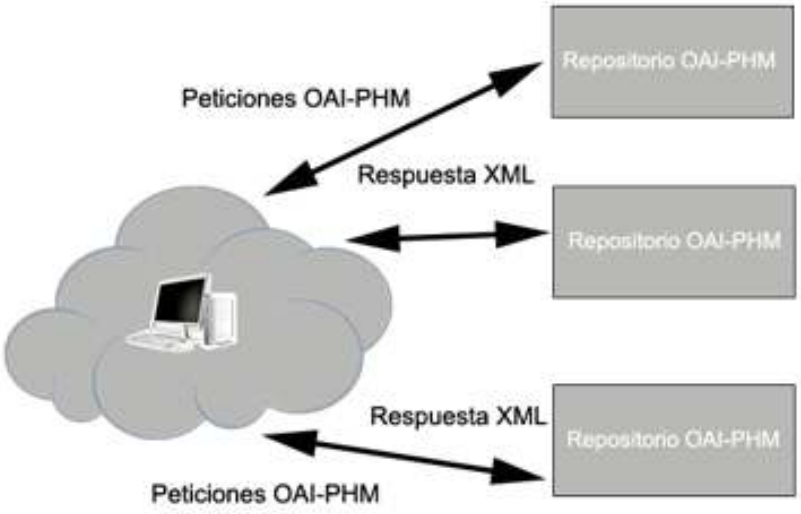

Figura 2 : Solicitudes y respuestas en el Protocolo OAI-PMH

El protocolo soporta múltiples formatos para expresar los metadatos, pero como mínimo los repositorios deben proveer los registros utilizando Dublin Core codificado en XML, además de este formato cada repositorio es libre de ofrecer los registros en otros formatos adicionales. Debido a que el protocolo OAI-PMH es muy extenso se recomienda su lectura completa desde su sitio web.

API de Harvester

Todo el proceso de obtención de información de los repositorios se realiza a través de un proceso batch, el cual se ejecuta en un día y hora especificado por las configuraciones de los Framework de Desarrollo implementados en el Sistema. En la actualidad existen muchas APIs de desarrollo que facilitan el trabajo de conexión mediante solicitudes OAI-PMH a los repositorios, la utilizada en el desarrollo del sistema se llama OAIHarvester2, la cual es un proyecto Java Open Source y soporta el protocolo OAI-PMH v1.2 y v2.0.

El proceso batch de Harvester realiza una serie de pasos para obtener toda la información necesaria de los repositorios y poder persistirla en la base de datos.

Paso1: El Sistema debe obtener mediante los Frameworks de desarrollo una lista de los repositorios de la tabla repositorios de la Base de Datos, en realidad lo que el Sistema obtiene en cada elemento de la lista es una dirección URL base de un repositorio, a la cual agregándole los verbos adecuados se puede realizar todo el proceso de Harvester de esa URL base.

Ejemplo de URL base

http://quod.lib.umich.edu/cgi/o/oai/oai

Paso 2: Luego el Sistema procede a realizar una solicitud al repositorio con la dirección URL base obtenida en conjunto con el verbo Identify del protocolo OAI-PMH. De la información obtenida en formato XML, el Sistema necesita saber el valor que tiene el atributo deletedRecord y guardarlo porque en etapas posteriores será necesario constatar dicho valor. Los posibles valores que puede adoptar el atributo deletedRecord son persistent (mantiene información sobre los registros), no (no mantiene información sobre los registros que se borran) y transient (no garantiza que se mantenga información sobre los registros que se borran), los cuales se proceden a explicar.

Solo se revisará la disponibilidad de aquellos registros cuyo repositorio utilice el atributo persistent en deletedRecord para poder borrar el registro de las tablas de la Base de Datos del Sistema, de forma que los mismos no se encuentren disponibles a la hora de que el usuario realice una solicitud de información con el objetivo de no devolverles información inexistente.

Paso 3: Un nuevo pasó que debe realizar el Sistema, es una solicitud con el verbo ListIdentifier junto con la URL base, este verbo tiene como restricción que se debe utilizar en conjunto con el verbo metadataPrefix el cual tiene un valor constante de oai_dc, porque se desea recuperar solo aquellos que se encuentren en el formato Dublin Core. La respuesta proporcionada por el repositorio será una lista de cabeceras de registros en los cuales se encuentra el identificador único de cada registro (forma de identificar unívocamente a un registro), el Sistema debe buscar en cada cabecera de respuesta XML el tag <identifier> correspondiente al identificador único en formato Dublín Core. Los valores encontrados se almacenarán en un Vector de identificadores únicos, por lo general en cada solicitud que se realiza al repositorio con el verbo ListIdentifier, el repositorio proporciona una lista incompleta de cabeceras de registros. Si el repositorio proporciona una lista con resumptionToken (indicador de que la lista obtenida es incompleta), se debe realizar tantas solicitudes como sean necesarias hasta obtener una lista completa. El vector de identificadores únicos tendrá tantos valores como la suma de registros de todas las listas incompletas, es decir que el número de identificadores únicos que tendrá el vector será igual al número de registros de la lista completa que debe devolver el repositorio.

Paso 4: La siguiente acción que debe realizar el Sistema es iterar sobre este vector de identificadores únicos y en cada iteración se debe realizar una solicitud al repositorio con la dirección URL base y el verbo GetRecord para obtener toda la información relevante a cada registro del Vector de identificadores. El identificador único permitirá al Sistema individualizar al registro que se encuentra en el repositorio, para poder realizar una solicitud OAI-PMH específica sobre dicho registro y obtener toda la información que necesite del mismo.

Solicitud GetRecord

http://arXiv.org/oai2?

URL base

verb=GetRecord&identifier=oai:arXiv.org:cs/0112017&metadataPrefix=oai_dc

verbo Identificador Único Metadato Dublin Core

Paso 5: El Sistema debe buscar en la respuesta que proporciona el registro los tags presentes en la Tabla 2 los cuáles se encuentran en formato Dublín Core y que representan la información que el Sistema extrae de los registros, luego procesa dicha información y por último persiste en sus Bases de Datos.

Tabla 2: Tags de Dublin Core a almacenar en Base de Datos.

Tags Dublín Core	Descripción
Titulo	El titulo correspondiente al registro analizado, es necesario porque se lo utiliza en el front-end del Sistema, para mostrar información referente al registro.
Creador	Nombre Descriptivo referente al creador del registro analizado, es necesario porque se lo utiliza en el front-end del Sistema.
URL del registro	URL correspondiente al registro analizado, la cual es utilizada en el front-end del Sistema para que el usuario pueda tener acceso al recurso original.
Descripción	Descripción del registro analizado y utilizado por el Sistema para lograr obtener los tags representativos de dicho registro, los cuales se utilizarán para realizar las asociaciones entre los tags.
Idioma	Idioma en el cual se encuentra el registro analizado, y utilizado por el Sistema para saber con qué idioma debe procesar la etiqueta descripción, para obtener los tags representativos de un registro
Publicador	Nombre descriptivo referente al publicador del registro analizado, utilizado en el front-end del Sistema.
Fecha(<date>) Fecha(<dateStamp>)	

Una condición que implementa el Sistema para poder persistir en la base de datos, es que los valores de todos los ítems en formato Dublín Core especificados en la Tabla 2 no deben estar vacíos, ya que es información necesaria para utilizar en el front-end del Sistema o para realizar procesos sobre otros campo de la Tabla 2. Por ejemplo, un elemento necesario para el Sistema y que se utiliza para mostrar en la pantalla que interactuará con el usuario es el item creador. Otro elemento necesario es el item descripción ya que el Sistema debe cerciorarse que el mismo no este vacío, para poder realizar el proceso de taggueo en la información que contiene ese campo. Si las validaciones existentes en el Sistema realizadas sobre un registro en particular se cumplen, el Sistema procede a persistir la información obtenida en sus tablas de la Base de Datos como se puede observar en la Figura 3.

Figura 3: Proceso de persistencia de datos procesados de un registro en Base de Datos.

Para realizar todo el proceso de recolección de información de los repositorios a través de solicitudes OAI-PMH el Sistema implementó las llamadas en back-end, es decir que en ningún momento se conecta a un browser o navegador web para realizar una solicitud, pero si es necesario contar con una conexión a Internet porque si bien no se conecta a los browsers, si debe conectarse con el repositorio a través de Internet y del protocolo HTTP en conjunto con el protocolo OAI-PMH. Las solicitudes a los repositorios las realiza a través de las clases Java y solicitudes las realiza a través del comando GET de HTTP.

Procesamiento de lenguaje natural

El Procesamiento del Lenguaje Natural nace como una subárea de la Inteligencia Artificial y la Lingüística. El Lenguaje natural es el medio que las personas utilizan de forma cotidiana para establecer la comunicación con el resto de las personas. La riqueza de los componentes semánticos proporciona a los lenguajes naturales un gran poder expresivo y un valor como una herramienta para razonamiento. Posee ciertos problemas que disminuyen la efectividad de los sistemas de recuperación de información en forma de texto:

· La variación lingüística: Es la posibilidad de utilizar diferentes palabras o expresiones para comunicar una misma idea.

· La ambigüedad lingüística: Se produce cuando una palabra o frase permite más de una interpretación.

El Lenguaje Natural tiene un alto grado de complejidad y se lo puede observar cuando se intenta recuperar información de un texto pretendiendo que esta satisfaga las necesidades de información del usuario que realiza una consulta. A continuación, se realiza una breve introducción de conceptos necesarios para lograr entender como el API de procesamiento de Lenguaje Natural funciona y lograr entender cómo se seleccionan las palabras representativas o tags de cada registro de un repositorio.

Lematización

La lematización es un proceso en cual se realiza la identificación de la forma canónica de una palabra, es decir su lema o dicho de otra manera trata de agrupar las diferentes formas de una palabra en un solo lema. El lema de una palabra comprende su forma básica más sus formas flexionadas es decir sus formas en plural, en femenino, conjugada en otros tiempos verbales, etc. Por ejemplo, la palabra informa podría ser el lema de información, informaciones, e informar.

Las técnicas de lematización son muy usadas cuando se necesita extraer información de un texto de forma automática, ya que para la recuperación de datos es muy útil reducir las variantes morfológicas que puede tener una palabra. Con estas técnicas se pretende ayudar al motor de búsqueda de tags para la recuperación de los mismos en la información que proveen los registros, y también al motor de búsqueda final con el cual interactuará el usuario.

Un lematizador es capaz de separar la raíz o lexema de una palabra de sus terminaciones, de tal forma que se puede asociar a una misma palabra las diferentes formas que se obtienen como resultado de la adición de sufijos. Habitualmente la lematización trabaja sobre los morfemas de género y número, a los cuales se pueden añadir otros como ser el caso de los

verbos de tiempo, persona, etc. Un lematizador se encarga de identificar los atributos finales que puede tener una palabra según su pertenencia a una determinada categoría gramatical para llegar al lema asociado a dicha palabra, el principal problema que surge es la ambigüedad que se genera en el momento de tratar de asociar una forma a una categoría gramatical determinada.

Con ello se introduce un nuevo concepto el cual es el sincretismo de formas, el cual se procede a explicar con el siguiente ejemplo: Se puede encontrar dos formas idénticas de la palabra "toque", la cual puede pertenecer a dos categorías gramaticales diferentes. Una es un sustantivo y la otra es una forma del verbo tocar.

La lematización puede realizarse automáticamente mediante programas de análisis morfológico, y existen diferentes grados de lematización posible: Se puede realizar una lematización puramente morfológica, o bien hacer una lematización sintáctica (en una oración) que tenga en cuenta el contexto en el que aparece la palabra.

Lematización Morfológica

La palabra "ama" tendría dos lemas: el sustantivo ama y el verbo amar.

Lematización Sintáctica

En "El ama de llaves abrió la puerta", ama es sustantivo. En "Carla ama a Juan", ama es verbo amar

Para poder hacer este tipo de lematización es necesario, por lo tanto, hacer un análisis sintáctico (Schmid Helmut). Con estas técnicas se reduce los problemas que tienen los lenguajes de procesamiento natural, mencionados anteriormente.

Etiquetado Gramatical

EL etiquetado gramatical (Part of Speech, POS tagging o POST) es el proceso de asignar (o etiquetar) a cada una de las palabras de un texto en una categoría gramatical. Este proceso se puede realizar en base a la definición de la palabra o el contexto en el cual la palabra aparece, como podría ser su relación con las palabras adyacentes en una frase, oración o en un párrafo. La gramática tradicional se encarga de clasificar las palabras basándose en ocho partes de la oración:

- Verbo.
- Sustantivo.
- Pronombre.
- Adjetivo.
- Adverbio.

· Preposición.

· Conjunción.

· Intersección.

Y las APIs de procesamiento de Lenguaje natural además utilizan:

· Artículo.

Muchas palabras pueden ser un sustantivo en una oración y un verbo o adjetivo en la siguiente (sincretismo).

Ejemplos de Etiquetado Gramatical

Sustantivo	Verbo
Juan	Trabaja.

Sustantivo	Verbo	Verbo
Juan	Esta	Trabajando.

Pronombre	Verbo	Sustantivo
Ella	ama	a los animales.

Sustantivo	Verbo	Adverbio	Sustantivo
Clara	habla	bien	el Inglés.

Sustantivo	Verbo	Sustantivo	Adjetivo
Clara	habla	Ingles	bien

Pronombre	Verbo	Preposición	Adjetivo	Sustantivo	Adverbio
Ella	Corre	a	La	Estación	rápido

La siguiente frase contiene todas las partes del etiquetado gramatical:

Pronombre	Conjunción	Adjetivo	Sustantivo	Verbo	Preposición	Sustantivo
ella	y	el joven	Juan	caminan	a	La escuela

API TreeTagger

El TreeTagger es una herramienta que permite realizar anotaciones sobre los textos de información, es decir sobre las partes de la oración (sustantivos, verbos, pronombres, etc.), fue desarrollado dentro del proyecto TC en el Institute for Computational Linguistics of the University ofStuttgart y para su proceso interno de taggueo hace uso de la Lematización y el Part of Spech.

Las notaciones que se utilizan para las etiquetas que se generan sobre la información que se desea analizar es tomada del proyecto Penn Treebank como se muestra en la Tabla 3 (para poder visualizar todas las notaciones, consultar la página del proyecto). El proyecto de Penn Treebank le agrega al texto natural una estructura lingüística, dependiendo de su gramática.

Tabla 3: Notación de etiquetas Penn Treebank

POS Tag	Descripción	Ejemplo
IN	preposition/subord. conj.	*in, of, like, after, whether*
JJ	Adjective	*green*
NN	noun, singular or mass	*table*
NNS	noun plural	*tables*
NP	proper noun, singular	*John*
NPS	proper noun, plural	*Vikings*
PP	personal pronoun	*I, he, it*
PP$	possessive pronoun	*my, his*

Para la selección de los tags de cada registro, se decidió seleccionar aquellas palabras que sean las más representativas en relación a la información que proveen. Para ello se decidió que luego de que el proceso de taggueo se haya realizado se debe seleccionar aquellas palabras que sean del tipo sustantivo ya que estas aportan más datos en relación a la información que se procesa. De acuerdo al idioma con el cual se procese la información del registro se obtendrán diferentes Pos Tag, los cuales nos indican de que tipo es cada una de las palabras. En este ejemplo el proceso de taggueo se realizó con el idioma en Ingles por ende se seleccionarán aquellas palabras que sean del tipo (Pos Tag):

· NN: Sustantivos singulares
· NNS: Sustantivos Plurales
· NP: Sustantivo Propio Singular
· NPS: Sustantivo Propio Plurales

Al igual que con el idioma ingles el API TreeTagger tiene una nueva tabla Pos Tag referente al idioma español, es decir los posibles valores que puede tomar una palabra analizada con el API de procesamiento de Lenguaje Natural son los siguientes:

· NC - Sustantivos Comunes
· NMEA – Sustantivos de Medidas
· NP – Sustantivos Propios

De acuerdo al idioma con el cual se realice el proceso de taggueo serán diferentes los valores Pos Tag que se deban seleccionar para ser los tags representativos de la información de un registro.

Selección de Tags

Una vez finalizado el proceso de búsqueda de tags en la información que los registros del repositorio brindan se debe proceder a definir cuántos de ellos serán considerados como tags representativos del registro que se esté analizando. A medida que se realizó el proceso en diferentes registros, se pudo observar que en la mayoría de las ocasiones la cantidad de Sustantivos que se obtienen del proceso son muchos y en general algunos son más representativos de la información analizada que otros. De la respuesta proporcionada por el proceso TreeTagger se deben seleccionar los sustantivos de tipo NC, NMEA y NP, y en general los Sustantivos más representativos de un registro serán los de tipo NC, NP que los de tipo NMEA, porque como se puede observar en la Tabla 4 la cual hace referencia a Sustantivos de tipos de medida (NMEA) no son muy representativos de la información que se procesa.

Tabla 4: Sustantivos de Medida

Sustantivos de Medida
docena
gramo
kilo
litro
centímetro
pulgada
decilitro

Los tags almacenados en la Base de Datos son los que se utilizan para realizar las asociaciones entre la información o entre clúster de información, y el resultado de dicha asociación de información es la que se encargará de recibir el front-end para mostrar los resultados de una búsqueda.

Se investigó sobre una metodología con la cual se pudiera realizar una selección apropiada, sobre qué tipos de Sustantivos representaran mejor a cada registro. Para ello se analizaron técnicas estadísticas y se seleccionó la media ponderada, pero esta técnica solo utiliza variables cuantitativas, mientras que el proceso de análisis de tags, devuelve variables cualitativas. Para lograr cumplir con los requerimientos de la media ponderada se utiliza las tablas de frecuencia, la cuales contienen el número de veces que aparece un determinado valor o en el caso planteado el número de veces que aparece una determinada palabra. Las tablas de frecuencias se representan por fi, y la suma de todas las frecuencias absolutas es igual al número total de datos, que se representa por N, para indicar resumidamente estas sumas se utiliza la letra griega Σ (sigma mayúscula) que se lee suma o sumatoria.

Ponderación de Valores

COMMONNOUNS= 35; Post Tag NC
MEASURENOUNS= 15; Post Tag NMEA
PROPERNOUNS= 50; Post Tag NP

Se establece una nueva restricción al Sistema, una vez finalizado el proceso de tabla de frecuencia y cálculo de ponderación, solo se almacenarán hasta 10 Tags representativos por cada registro, con los cuales luego se procederá a realizar el clúster de información para proveer los datos necesarios al front-end. La restricción en cuestión es una restricción de performance porque si se colocan más tags, al momento en el que el usuario realice una búsqueda se realizarán las asociaciones entre los tags almacenados en la Base de Datos y la información devuelta para ser procesada por el front-end es muy grande y por ende la visualización de los mismos es demasiado lenta.

Visualización de datos

La información como se almacena en la actualidad no facilita su recuperación y manipulación de una manera automatizada, como se planteó en el desarrollo del Sistema. Es decir que una información existente en la Web como podría ser un documento del área científica, del área de física, etc., no se relaciona con otros documentos de la misma área por intermedio de palabras claves que las representen y que faciliten de ese modo una asociación entre la información para realizar las búsquedas en la Web. Los repositorios de información son de gran ayuda porque permiten obtener la información almacenada en ellos en un formato específico y de tal forma permitir al Sistema poder procesar dicha información.

La Web fue realizada principalmente para que la información contenida en la misma sea de consumo humano, se pretende que con el desarrollo del Sistema se provea de una interfaz en la Web que provea datos que sean procesados por las máquinas, la cual es denominada Web Semántica (Peis, Herrera-Viedma y Hassan, 2003) y se puede observar en la Figura 4 , éste es un proceso al que se describe como Bringing the Web to its full potential (Dieter; Hendler, et al., 2005). En la cual se puede observar las asociaciones existentes entre diferentes áreas con la finalidad de obtener información más completa referente a un tópico de búsqueda.

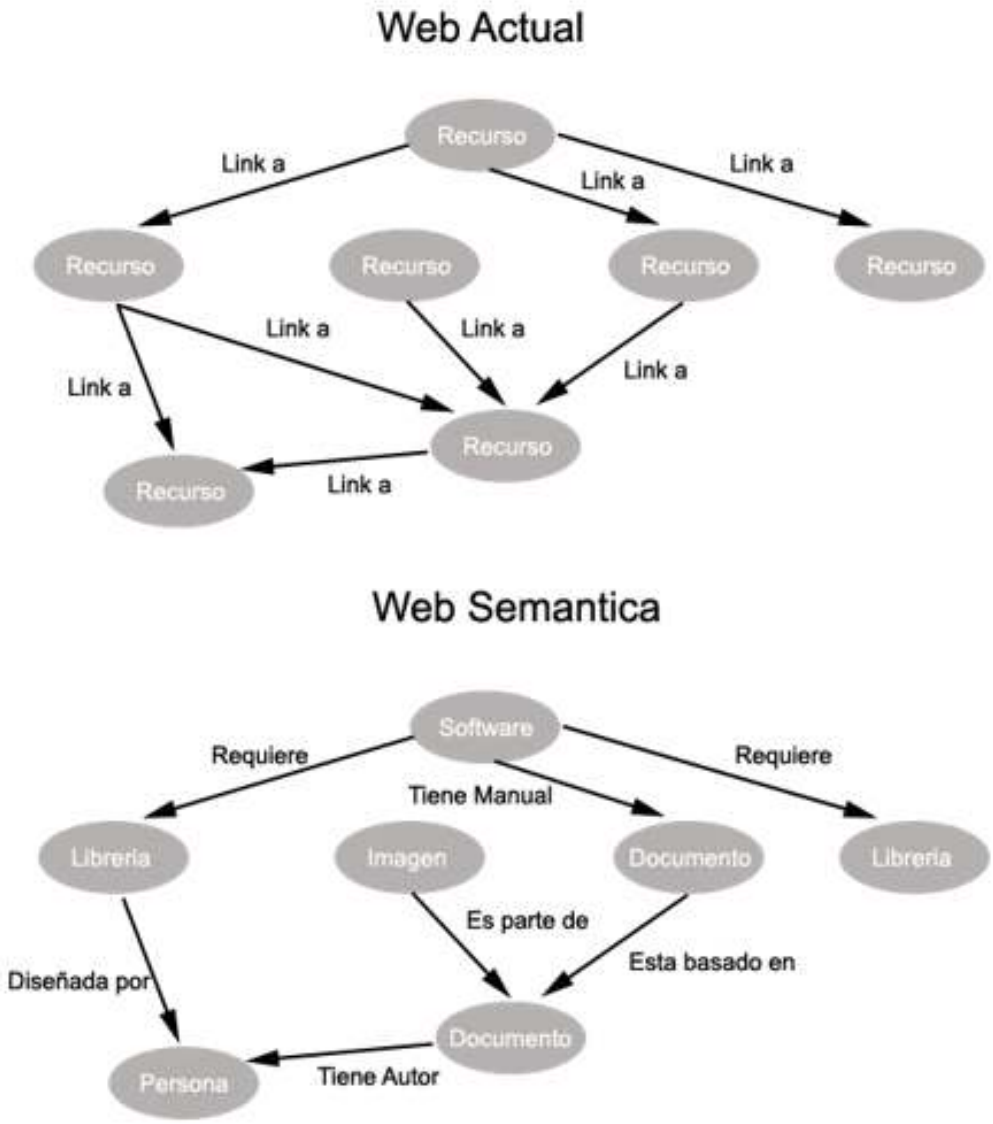

Figura 4: Ejemplo Web Semántica

Una de las principales características que diferencian al ser humano del resto de las especies es la capacidad que tiene para interiorizar y procesar información, transformándola en conocimiento. En los textos de psicología cognitiva, el ser humano es descripto como un “informivoro”, del mismo modo que un carnívoro se nutre de carne la información es el alimento de la cognición del ser humano y su consumo es una necesidad básica para la actividad diaria y su adaptación al entorno. Como se mencionó los repositorios nos proporcionan cada vez mayor información, pero la capacidad de los seres humanos para percibirla, interiorizarla y procesarla permanece estática en términos cuantitativos, ya que se encuentra condicionada por la propia biología del ser humano.

Como consecuencia de ello, las personas tienden a economizar constantemente la atención, discriminado de esa forma toda aquella información que resulte irrelevante para sus necesidades e intereses, con el objetivo de evitar la saturación de información.
El campo de la visualización es muy amplio y el sistema hace uso de la visualización de la información la cual hace mención a la visualización de datos abstractos. La visualización de la información es entendida solo sobre grandes volúmenes de datos, es decir aquellos cuyas estructuras subyacentes únicamente pueden ser extraídas algorítmicamente y no a través del análisis y reflexión subjetiva del diseñador de la visualización.
Ante el incremento acelerado de la cantidad de información que se hace presente en la Web, se hace casi prácticamente imposible para una persona poder extraer conclusiones, tendencias y patrones a partir de los datos crudos. Es aquí donde el concepto de visualización hace su aporte significativo, y la exploración de distintos conjuntos de datos se beneficia enormemente si cuenta con un soporte adecuado de visualización. La visualización de la información es una tarea que realiza el Sistema, y el mismo trata de cumplir con el rol de comunicador visual, es decir que se encarga de transformarlos datos abstractos de la realidad en mensajes visibles permitiendo a los usuarios del Sistema que vean con sus propios ojos, datos y fenómenos que son directamente inaccesibles, y que por tanto comprendan la información que se encuentra oculta (Costa, 1998).
Uno de los campos de la Visualización de la Información es la Visualización de Grafos que representa el problema de visualizar la información construyendo representaciones visuales geométricas de grafos o redes. Se implementó en el Sistema la capacidad para mostrar gran cantidad de información, mediante visualización de grafos.

Conclusiones

El propósito del desarrollo del prototipo estuvo basado en que no existe en la web un Sistema con las características que se planteó en este documento. Si bien las búsquedas que se realizan hoy en día en los buscadores de la web, son perfectamente funcionales y nos muestran los resultados asociados a la búsqueda en cuestión, con este desarrollo tratamos de ir un poco más adelante tanto en la forma en la cual se realizan las búsquedas como así también en la forma de representar los resultados al usuario.
Para lograr este objetivo, se inició realizando una arquitectura la cual fuera lo suficientemente estable para poder ir agregando todos los conceptos que se plantearon e incorporando progresivamente los frameworks que ayudaron a realizar el desarrollo del prototipo de una manera más rápida y precisa. Con algunos conceptos estaba claro que herramientas utilizar gracias a la guía del tutor, en otros se tuvo que analizar, probar las diferentes opciones que se encontraron durante la investigación para ver cuál de ellas se adecuaba mejor a las necesidades del desarrollo.

Una vez terminado el proceso batch del prototipo encargado de recolectar información de los repositorios, aplicarle los algoritmos necesarios para poder clasificar la información recolectada y almacenarla en la base de datos. Se encaró el nuevo desafío, el cual es un área de la inteligencia artificial, el de crear una nueva "forma" de mostrar los datos a los usuarios mediante el concepto de "Visualización de la Información". Actualmente en la web se puede encontrar muchas herramientas sobre como representar la información para que sea mejor comprendida por el usuario, para ello se probó diferentes herramientas tantos comerciales como gratuitas y se seleccionó la librería D3 completamente desarrollado en JavaScript.

Además de ser la que más se adecuaba a las necesidades de visualización, es la que proporcionaba mayor documentación sobre sus funcionalidades y en base a esta librería se implementó la nueva visualización de información para el usuario. Fue necesario mucho estudio, sobre como accede la librería a los componentes que se dibujan en pantalla, es decir a los componentes del documento HTML que se presenta al usuario. Porque los componentes que representan a la información y la forma en la que se conectan entre ellos se hace de manera dinámica.

No fue una tarea sencilla debido a que la librería tiene sus propios métodos de acceso al DOM (Modelo de Documentos de Datos) del documento HTML, y fue necesario realizar modificaciones constantes tanto a la librería como al componente que se desarrolló para poder generar la vista del usuario. Por lejos el componente más complicado de todo el desarrollo del prototipo fue la generación de la vista, ya que incluyo mucho tiempo de análisis, investigación y pruebas para lograr obtener la vista deseada.

Si bien se logró alcanzar el objetivo propuesto y desarrollar un prototipo de sistema el cual fuera capaz de obtener información de los repositorios, clasificarlas y por último mostrarla al usuario, durante las diferentes etapas de investigación/desarrollo se fueron encontrando acotaciones que deben ser consideradas como una futura mejora del prototipo.

· Mejora en los tiempos de respuestas de las querys encargadas de buscar los resultados coincidentes con los criterios de búsqueda seleccionados por el usuario. Los tiempos de respuestas por lo general no son prolongados, pero se debería buscar la forma de optimizarlos, una posible solución podría ser agregar índices a las tablas de la Base de Datos.

· Adicionar más idiomas en los criterios de búsqueda que puede seleccionar el usuario, si bien solo se utilizaron el idioma español y el inglés, la arquitectura del prototipo como así también el API de procesamiento de Lenguaje natural soporta muchos más idiomas. Con lo cual aumentaría considerablemente el volumen de información que se almacena en la Base de Datos cuando se realiza el proceso de selección de palabras claves, lo que conllevaría a que el punto anterior sea prioritario en una futura mejora.

· Mejorar los algoritmos de búsqueda que actualmente se desarrollaron para el prototipo, si bien un ítem es mejorar los tiempos de búsqueda en la Base de Datos, una vez que la información vuelve al sistema es sometida a un proceso que relaciona los datos obtenidos y los prepara con un formato adecuado para que la vista lo pueda procesar.

· Mejoras en los algoritmos de procesamiento del Lenguaje Natural, actualmente se utiliza el API TreeTagger para obtener las palabras claves de cada repositorio. Pero con la investigación realizada se observó una nueva API que en "teoría" es mucho más potente y proporciona muchas más funcionalidades que el API utilizada. Se denomina NLTK (Natural Languaje Toolkit) y se encuentra implementada en el lenguaje de programación python y es una mejora sustancial que se podría incorporar al sistema en futuras versiones.

Durante todo el proceso de análisis y desarrollo del prototipo se fueron solucionando diferentes problemas que surgieron a la hora de incorporar todos los frameworks/APIs para que funcionaran como una sola unidad funcional y poder dar por terminado el desarrollo del prototipo de Sistema.

Bibliografía

Hassan, Yusef: Núñez Peña, Ana. (2005). Diseño de Arquitecturas de Información: Descripción y Clasificación. Revista no solo Usabilidad sobre personas, diseño y tecnología. Extraído el 14 de enero del 2012, de
www.nosolousabiliad.com/articulos/descripcion_y_clasificacion.htm

Hassan, Yusef. (2006). Visualización y Recuperación de Información. II Encuentro de Ciencias y Tecnología de Documentación e Información. Escuela Superior de Estudios Industriales y de gestión. Universidad de Granada.

Cañada, Javier. (2006). Tipologías y estilos en el etiquetado social. Extraído el 14 de enero del 2012, de www.herrero.ugr.s/gbd/docs/tagging.pdf

Anderson, James; Perez Carballo, Jose (2001). La naturaleza de la indexación: Como los seres humanos y maquinas analizan los mensajes y textos para su recuperación. Parte 2: Maquina de indexación y la asignación de recursos humanos contra la máquina. Revista Procesamiento y Gestión de la Información, Vol. 37, N°2.

Perkins, Jacob (2010). Procesamiento de texto con Phyton 2.0 NLTK Cookbook. Birmingham: Packt Publishing LTD.

Ware, Colin (2004). Information Visualization: Perception for Design. New Hampshire: Morgan Kaufmann.

Ware, Colin (2008). Visual Thinking for Design. New Hampshire: Morgan Kaufman.

Tufte, E.R. (1983). The Visual Display of Quantitative Information. Cheshire: CT Graphics

Press.
Tufte, E.R. (1997). Visual Explanations: Images and Quantities, Evidence and Narrative. Cheshire: CT Graphics Press.
Carlis, John; Konstan, Joseph (1998). Interactive visualization of serial periodic data. In ACM Symposium on User Interface Software and Technology. New York: ACM Press.
Card, Stuart; Mackinlay, Jock; Shneiderman, Ben. (1999). Readings in Information Visualization: Using Vision to Think. San Francisco: Morgan-Kaufmann.
Dublin Core Metadata Initiative. Extraido el 15 de enero del 2012, de http://www.hipertexto.info/documentos/dublin_core.htm.
David Gourley; Brian Totty. (2002). HTTP: The Definitive Guide. UniteStates of América: O'Railly.
Peis, Eduardo; Herrera-Viedma, Enrique; Hassan, Yusef (2003). Herrera. Análisis de la web semántica: estado actual y requisitos futuros. Revista El profesional de la Información, Vol. 12, N°5
Joan Costa (1998). La esquemática: Visualizar la información. Argentina: Paidós.