Generación de descripciones de Web Semántica en una red de ONG

Tutores:

Dra. Dina Wonsever
Ing. Pablo Accuosto
Ing. Diego Garat

Integrantes:

Natalia Chiaro
Pablo Damonte

Instituto de Computación
Facultad de Ingeniería
Universidad de la República
2004 – 2005

Resumen

En los últimos años ha surgido interés en transformar la *World-Wide Web*, actualmente un enorme repositorio de información, en una gran base de conocimientos. Extender la *Web* con esta capacidad de almacenamiento e inferencia daría paso a lo que se ha dado en denominar *Web Semántica*.

Para que el conocimiento almacenado en la *Web* pueda ser efectivamente recuperado y utilizado en forma automática es necesario enriquecer las páginas con *metadatos* que permitan definir conceptos y relaciones en un dominio específico. Estos conceptos y relaciones se expresan mediante el uso de un conjunto controlado de términos que se denomina *ontología*. Por lo tanto, la generación de ontologías en distintos dominios es esencial para posibilitar la construcción de la *Web Semántica*.

En este proyecto se realizó un análisis del estado del arte para proponer herramientas y técnicas eficaces que permitan, con la menor intervención humana posible, generar una ontología partiendo de un sitio *Web* en un dominio dado. Para esto se analizaron técnicas de extracción de información, lenguajes de definición de *ontologías* y herramientas que permitan su generación y posterior explotación.

La solución propuesta parte de una ontología mínima, construida manualmente, que se completa con entidades nombradas y relaciones identificadas automáticamente mediante técnicas de extracción de información aplicadas en el sitio *Web* escogido para el estudio.

Conjuntamente con la generación de la ontología se realizó un prototipo que muestra en forma práctica los beneficios de disponer de la misma.

Una evaluación del grado de reconocimiento y precisión en la recuperación de documentos basándose en la ontología generada permite estimar que las técnicas propuestas pueden constituir un aporte relevante para la generación semiautomática de *ontologías* a partir de páginas *Web*.

Palabras claves: Web semántica, metadatos, ontología, extracción de información.

Índice

Capítulo 1 Introducción

La "*Web Semántica*", propuesta inicialmente por Tim Berners-Lee [1], es actualmente una iniciativa del *World Wide Web Consortium* (W3C) [2]. Concebida como la siguiente etapa en el desarrollo de la *Web*, entiende que ésta sólo puede alcanzar su pleno potencial si el conocimiento contenido en ella, que actualmente es "entendible" únicamente con intervención humana, puede ser compartido y procesado por herramientas automáticas. Para lograr este fin, es necesario enriquecer y estructurar la información disponible actualmente en la *Web*, lo cual implica acordar formatos para la representación de este conocimiento, así como mecanismos que posibiliten una utilización eficaz del mismo por parte de aplicaciones desarrolladas de forma independiente.

Para que pueda existir un procesamiento semántico de los documentos en un escenario *Web*, idea fundamental de la *Web Semántica*, es necesario tener en cuenta tres elementos fundamentales. El primero de ellos se basa en la idea de tener agentes de software con la capacidad de procesar automáticamente documentos, realizando para ello un análisis semántico del contenido de los mismos. El segundo concepto es la existencia de un modelo conceptual que describa los rasgos característicos de un dominio dado, es decir lo que se conoce como una ontología del dominio. El último elemento, consiste en la existencia de *metadatos*, es decir, aquella información asociada a los documentos que describe el contenido semántico de los mismos.

La calidad y la correcta utilización de estos elementos, posibilitará a los usuarios un mayor dominio a la hora de localizar y entender los datos de las páginas pertenecientes a un dominio específico.

El presente proyecto está orientado a la generación semiautomática de ontologías basadas en páginas *Web* pertenecientes a un dominio específico.

1.1 Motivación y objetivos

Choike[1] es un portal destinado a mejorar la visibilidad de los contenidos producidos por las organizaciones no gubernamentales (*ONG*) del sur, además de ser una plataforma donde las *ONG* pueden difundir su trabajo y a su vez alimentarse de diversas fuentes de información organizadas desde la perspectiva de la sociedad civil del sur.

Actualmente *Choike* ofrece:

- Un directorio de *ONG* organizado por temas. Existe actualmente un conjunto de unos 20 temas, organizados en dos niveles.
- *Un buscador que permite rastrear información en los sitios de las ONG*. Ésta herramienta permite buscar en el conjunto acotado de páginas *Web* que por su calidad y relevancia forman parte del directorio.

Actualmente, el portal carece de un buscador avanzado que permita, por ejemplo, encontrar las páginas que tratan de temas similares a los tratados en otra página o encontrar los términos más mencionados de una página.

[1] www.choike.org

Para mejorar el acceso a las páginas y permitir búsquedas más avanzadas a las permitidas actualmente, es que se pretende la generación de *metadatos* de las páginas del sitio *Choike*, y de una ontología específica para dicho sitio.

El objetivo principal del presente proyecto es la generación de *metadatos* según una especificación estándar de *Web Semántica* y la organización de descriptores en una ontología. Con tal fin, se analizarán y evaluarán distintas técnicas para la asociación de descriptores a páginas utilizando técnicas de aprendizaje automático y la generación semiautomática de la ontología "aprendiendo" términos y asociaciones a partir de las páginas *Web,* que llevarán a la construcción de una solución al problema planteado.

Para comprobar la viabilidad de dicha solución, se llevara a cabo la construcción de un prototipo que a partir de un conjunto de páginas (informes) de *Choike* genera una ontología específica.

1.2 Alcance

Los resultados esperados son el estudio sintético del estado del arte en herramientas para generación de *metadatos* y ontologías, la propuesta de una herramienta de software para su generación, la evaluación de los resultados en el contexto de la red *Choike* y, adicionalmente, el desarrollo de un prototipo que muestre las ventajas de utilizar ontologías para la búsqueda de información y otras consultas útiles para los usuarios.

1.3 Organización del documento

El presente informe se estructura en seis capítulos que brindan una visión global de los aspectos más destacados del proyecto en el que se enmarca. A continuación se realiza una breve reseña de cada uno de los capítulos.

En el capitulo 1, se analiza y estudia el actual estado del arte de las herramientas de definición y gestión de ontologías. Las herramientas implementadas para la extracción de información y generación semiautomática de una ontología se describen en el capitulo 3, mientras que en el capitulo 4 se presenta una evaluación de los resultados obtenidos en dicha generación. Finalmente en el capitulo 5 se brinda una descripción del prototipo construido para mostrar algunas de las potenciales funcionalidades que se obtienen al disponer de una ontología específica. En el capitulo 6 se presentan las conclusiones y trabajos futuros.

Capítulo 2 Ontologías

2.1 Introducción

En este capítulo se pretende dar una idea general del estado del arte de uno de los conceptos principales involucrados en la llamada *Web Semántica*, las *ontologías*. Para ello, se describirán las distintas concepciones del término a lo largo del tiempo, enfocándose posteriormente en la diversidad de sus usos prácticos. Por otra parte, se analizan diferentes lenguajes que permiten su definición y utilización, y las herramientas actualmente existentes para su gestión.

2.2 Ontologías

El término *ontología*, ha sido ampliamente usado en el correr de los años. En *el año* 1721, el diccionario de la editorial Estadounidense Merriam-*Webster*, provee dos definiciones:

1 – una rama de la metafísica concerniente a la naturaleza y relaciones de ser.
2 – una teoría particular sobre la naturaleza de ser de los tipos existentes.

Estas definiciones, ofrecen una noción abstracta y filosófica de *ontología*.

Ya en 1900, la noción de una *ontología* formal había sido distinguida en la lógica formal por el filósofo *Husserl*. A pesar de que las *ontologías* (incluso las *ontologías* formales) tienen una larga historia, siguen siendo tema de interés académico entre filósofos, lingüistas, bibliotecarios e investigadores de la representación del conocimiento.

Gruber [3] por su parte, dio su visión diciendo que "una *ontología* es una especificación explícita de una conceptualización".

Para *Taylor* [4], las *ontologías,* que en el campo de la recuperación de información suponen un avance en la interrelación entre las personas y las computadoras, pueden ser no lingüísticas (empleadas para la creación de agentes inteligentes) y lingüísticas, estando vinculadas con aspectos gramaticales, semánticos y sintácticos. En unos casos, estas ontologías lingüísticas se reducen a una lista jerárquica de términos de un área específica y en otros, son vocabularios controlados, categorizados, que incluyen un análisis semántico de palabras para su posterior categorización y vinculación.

Las *ontologías* han estado ganando interés y aceptación en el área de la informática (además de la filosófica). Las personas (y agentes computacionales) generalmente tienen una noción o conceptualización del significado de los términos. Los programas de software a veces proveen una especificación de las entradas y salidas de un programa, lo que podrían usarse como una especificación del programa. Similarmente, una *ontología* puede ser usada para proveer una especificación concreta de los nombres y significados de los términos. Dentro de la línea de pensamiento dónde una *ontología* es vista como una especificación de la conceptualización de un término, hay todavía varias interpretaciones potenciales.

Una de las nociones más simples de una posible *ontología* puede ser un vocabulario controlado (una lista finita de términos). Un catálogo o glosario es un ejemplo de esta categoría.

Para considerar el impacto de aplicar una *ontología*, es conveniente considerar las *ontologías* de dos formas distintas: las simples, y las más sofisticadas.

Una *ontología* simple, no es costosa de construir, y además se encuentra disponible de varias formas – algunas existen como *freeware* en la *Web* y también como información de la estructura interna de una organización dentro de compañías, universidades, etc. Algunos esfuerzos colectivos existentes como DMOZ[2], están generando grandes *ontologías* simples.

Dentro de los usos prácticos de una *ontología* simple podemos encontrar:

- Vocabulario controlado
- Sitio de una organización y soporte de navegación
- Soporte para navegadores
- Soporte de búsqueda
- Soporte para la desambiguación.

Una *ontología* estructurada es prácticamente una simple categorización para ser usada en aplicaciones. Ésta comienza con tener una estructura, sin embargo puede proveer de más poder a las aplicaciones.

Algunos de sus usos prácticos son:

- Chequeo de consistencia
- Completitud
- Soporte de interoperabilidad
- Soporte para la validación y prueba de verificación
- Soporte de configuración
- Soporte estructurado, comparativo y búsqueda personalizada.
- Aprovechar la especialización/generalización de información.

Las ontologías proveen una manera de representar y compartir el conocimiento utilizando un vocabulario común, permitiendo manejar un formato para intercambiar ese conocimiento, facilitando su reutilización. Se han clasificado de acuerdo al tipo de conceptualización que abarcan, destacándose aquellas que dan una idea de vocabulario común (terminológicas), las que ofrecen una idea de almacenamiento estándar de la información (de información) y las que se utilizan para el modelado del conocimiento.

Si se desea obtener una clasificación por el alcance que pueden tomar, las *ontologías* se dividen en tres grupos: las de dominio, que son específicas para un dominio concreto; las de tareas, que representan las tareas que se pueden realizar en un dominio específico y las generales, representando datos generales y no específicos de un dominio. Finalmente, algunos autores plantean una clasificación según la usabilidad y reusabilidad.

Las *ontologías* se pueden ver como esquemas de *metadatos*, que proveen un vocabulario controlado de términos, cada uno de ellos definido explícitamente con una semántica procesable por una máquina.

Dentro del escenario *Web,* las ontologías pueden verse como un espectro detallado de especificación, donde para completar la tarea de definir las teorías de dominios, se une la necesidad de estandarización de los diversos lenguajes desarrollados para su generación y manipulación los cuales se verán más adelante.

[2] DMOZ – (www.dmoz.com) por ejemplo, consta de más de 35.000 editores voluntarios, y tiene más de 360.000 categorías o clases taxonómicas

La falta de una metodología estándar para la creación de una *ontología*, ha motivado la búsqueda de una guía para un modelo de proceso que permita realizar esta tarea de forma clara, objetiva, completa, correcta, extensible y modularizada. [5]

Por último, una de las principales dificultades que se debe enfrentar al crear una *ontología* es que para ser capaz de desarrollarla y mantenerla manualmente, se debe ser experto en el tema y en el dominio, requiriéndose un gran esfuerzo para que permanezca consistente y coherente. Por este motivo, se han efectuado variados estudios enfocados a automatizar o semi-automatizar la generación de las mismas, por ejemplo a través de métodos orientados a tener elementos que conlleven a un entrenamiento previo para el procesamiento.

2.3 Descripción de lenguajes para definición de ontologías

Para que la *Web Semántica* se vuelva efectiva, el intercambio de *metadatos* debe ser realizado teniendo en cuenta los aspectos relevantes a la interoperabilidad semántica, sintáctica y estructural.

La comprensión de cada uno de los descriptores de las distintas fuentes y sus relaciones es posible gracias a la interoperabilidad semántica, lograda por el uso de vocabulario específico, ontologías y estándares para *metadatos*. Los *metadatos*, nos permiten tener información extra de los datos, estando su diseño muchas veces influenciados por una ontología.

La Figura 1 muestra la relación entre las ontologías, *metadatos* y los propios datos.

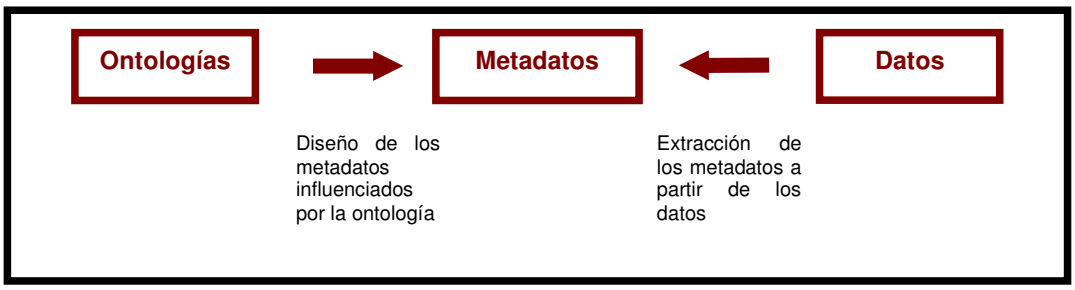

Figura 1 - Relaciones

Existe una gran diversidad de lenguajes de descripción de ontologías, cada uno con sus características específicas, contando con diferentes niveles de expresividad a la hora de describir los elementos básicos pertenecientes a una ontología: los conceptos, las relaciones, las funciones, los axiomas y las instancias.

Algunos de los lenguajes existentes brindan un grado de expresividad limitado al no permitir asociar semánticamente sus etiquetas, como es el caso de *XML* (*Extended Markup Language*).

SHOE (*Simple HTML Ontology Extensions)* [6][7] y *OIL* (*Ontology Inference Layer*) [8] avanzan un poco más, y permiten definir un vocabulario al cual se le asocia un significado que es entendido a nivel de máquina, pero carecen de mecanismos para expresar negaciones o disyunciones. En un nivel más avanzado a estos, encontramos a *RDF Schema* [9], quien provee de mecanismos para la declaración de propiedades referidas a los recursos, relaciones y clases.

Siguiendo el camino de *RDF Schema*, pero permitiendo definir relaciones más complejas entre las entidades, como ser, mecanismos para limitar las propiedades de las clases respecto al número y al tipo, o inferencias para determinar la clase del objeto a partir de sus propiedades y poseer un modelo de herencia bien definido, es que se encuentra *OWL* [10]. Pensado para ser usado cuando la información contenida es procesada por las aplicaciones y no por humanos, OWL es parte del grupo creciente de recomendaciones de *W3C* relacionada a la *Web Semántica*.

Existen lenguajes que además permiten la representación de conceptos organizados en taxonomías, relaciones o axiomas de lógica de primer orden como son los casos de *CKML* (*Conceptual Knowledge Markup Language*) [11], *OML* (*Ontology Markup Language*) y *XOL* [12]. Otros van un poco más allá de la lógica anterior, y permiten expresar los conceptos del mundo real, como es el caso de *Cycl* [13].

Por otro lado, tenemos otras iniciativas que apuntan a proporcionar un lenguaje y un conjunto de herramientas que habilitan la transformación de la *Web* de una plataforma que presenta información a una plataforma que entiende y razona información, como es el caso de *DARPA Agent Markup Language* (*DAML*)[14] o *DAML+OIL* [15]. Dentro de este conjunto también encontramos a *Knowledge Interchange Format* (*KIF*) [16] que posee una semántica declarativa en la que el significado de las expresiones en una representación puede ser entendido sin necesidad de recurrir a un intérprete para la manipulación de estas expresiones.

A su vez, existen lenguajes de propósito específico, como ser *EbXML* [17], pensado para el comercio electrónico o *F-Logic*, que es un lenguaje de especificación de bases de datos deductivo y orientado a objetos, combinando la semántica declarativa y expresividad de los lenguajes de bases de datos deductivos con las ricas capacidades de modelado de datos soportadas por el modelo orientado a objetos [18].

Grail [19], en cambio, es utilizado principalmente para producir modelos de terminología médica, permitiendo jerarquía de roles y formas restringidas de inclusión conceptual de axiomas. Incluye además funcionalidades para soportar operaciones léxicas en diferentes idiomas.

Para el ambiente de los desarrolladores de aplicación, existe *OCML* (*Operations Conceptual Modelling Language*) [20], que permite proporcionar restricciones y dependencias de bibliotecas y herramientas.

Por último, están los lenguajes que aunque su concepción no fue pensada para modelar ontologías, permiten hacerlo. Tal es el caso de *UML* (*Unified Modeling Language* [21] que fue pensado para la especificación, construcción y visualización de un sistema de software. *UML* puede ser utilizado también como lenguaje de modelado de ontologías brindando la ventaja de soportar la representación gráfica de las mismas, sin embargo tiene una menor expresividad que muchos de los otros lenguajes, los cuales fueron desarrollados específicamente para esta función: UML se ajusta más al diseño de ontologías que serán procesadas por agentes humanos que por agentes automáticos.

Luego de hacer esta reseña general sobre distintos lenguajes de definición de ontologías, puede observarse fácilmente la gran heterogeneidad y diversidad de opciones que se tiene a la hora de elegir un lenguaje. Es importante la aclaración de que no todos los lenguajes son adecuados o directamente aplicables a cualquier contexto, habiendo lenguajes específicos para un área de aplicación y otros de propósito más general.

Una aspecto importante a considerar, es el hecho de que muchos lenguajes, son extensiones de otros (esto puede apreciarse en la Figura 2), habiendo en muchos casos compatibilidad por lo menos en una dirección entre los distintos lenguajes.

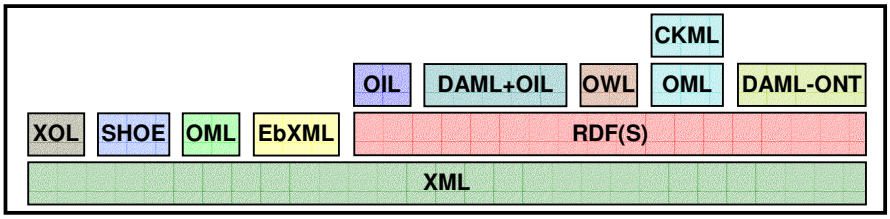

Figura 2 - Jerarquía de lenguajes en la Web Semántica

En la evaluación de los lenguajes antes mencionados, se hizo principal hincapié en la variedad de posibilidades que estos brindaban, como ser la de representar conceptos y relaciones, la facilidad de uso, etc., así también como el soporte que a estos le daban las distintas herramientas que actualmente se encuentran disponibles en el mercado, su grado de estandarización y la aceptación que hay de los mismos.

Se puede encontrar información adicional, sobre los distintos lenguajes de definición de ontologías y sus respectivas características, en el Anexo 1 – Ontologías.

2.4 Implicaciones y necesidades relacionadas con ontologías

Una importante consideración a tomarse en cuenta en referencia a las ontologías es el estudio de herramientas para el mantenimiento de las mismas a lo largo del tiempo. Si la ontología será mantenida por expertos en la materia (usuarios con escasos o nulos conocimientos del lenguaje en que se definió la ontología) y no por expertos de conocimiento (usuarios con amplio dominio de los distintos lenguajes de definición de ontologías y demás), probablemente se necesiten algunas herramientas ontológicas que permitan, por ejemplo, editarlas. Hay varias herramientas de las ontologías simples disponibles comercialmente. Algunas compañías de recuperación de información utilizan editores simples para generar y "navegar" jerarquías de generalización simples.

Las herramientas son variadas, por ejemplo *Ontolingua* [22] y *Chimaera* [23] de la *Universidad de Stanford*, *Oiled* [24][25] de la *Universidad de Manchester* y *Protégé* [26] de *Stanford Medical Informatics*, sólo para nombrar algunos.

Algunas compañías con necesidades de ontologías extensas como *VerticalNet* [27] tienen o están desarrollando sus propias herramientas para construir ontologías que satisfagan las necesidades de los usuarios más sofisticados. Dichas herramientas se construyen principalmente en base a distintos prototipos existentes en el mercado, resultados de diversas investigaciones.

Al seleccionar un lenguaje para usar o construir un ambiente de ontología, hay varios problemas que deben ser considerados:

- *Colaboración y soporte distribuido.* Algunos ambientes de ontologías permiten a los usuarios compartir una sesión. Esto puede ser particularmente útil para la depuración, volviéndose mucho más común con arquitecturas cliente/servidor. La colaboración puede requerir controles de concurrencia, bloqueos, y un tipo de sistema de versionado y permisos. Como ejemplo de herramienta que soporta esta funcionalidad podemos mencionar a *Ontolingua*.

- *Interconectividad de plataformas.* Cuando se embeben aplicaciones en plataformas más complejas, se torna importante para ambientes poder leer y escribir formatos compatibles, para que de esta manera puedan ser integrados con ambientes de hardware/software múltiples, etc. Las aplicaciones basadas en *Java* proporcionan una aproximación conveniente a este problema, pero también otros sistemas soportan múltiples entradas y formatos de salida, entienden estándares comunes, y proporcionan servicios de traducción y mapeo que pueden ayudar a esta tarea.

- *Escala.* Muchas aplicaciones de ontologías pueden necesitar ser escaladas y es importante mirar la escala en términos del tamaño de las ontologías así como el número de usuarios simultáneos.

- *Versionado.* Cuando las aplicaciones tienen una larga vida útil y se implantan en ambientes diferentes, se hace importante la necesidad de soportar y controlar diferentes versiones de ontologías.

- *Seguridad.* Algunas aplicaciones tendrán diferentes necesidades para el acceso a distintas secciones de la ontología, por lo tanto es importante tener un ambiente que sea capaz de exponer secciones de la ontología basado en un modelo de seguridad.

- *Análisis.* Se esperan ambientes que soporten la adquisición, evolución, y mantenimiento de ontologías. El soporte de análisis que pueda enfocar la atención del usuario en áreas probables de modificación pueden ser bastante útiles. El ambiente de ontologías *Chimaera*, por ejemplo, soporta varias pruebas de diagnóstico apuntadas a ayudar a los usuarios a identificar las ontologías incorrectas así como los posibles problemas.

- *Ciclo de vida.* Cuando las ontologías crecen más y más, se esperaría que los diseñadores de la aplicación puedan mantener las ontologías durante su ciclo de vida. Además, constantemente pueden estar uniendo nuevas ontologías en su sistema como también interconectar sus aplicaciones con sistemas más diversos.

- *Facilidad de uso.* Aun cuando un ambiente tiene todo lo que un diseñador de la aplicación puede necesitar, es difícil para el usuario decidir cómo utilizar las funcionalidades que éste brinda. Es de vital importancia entonces la existencia de materiales para entrenamiento, guías didácticas, soporte del modelo conceptual, herramientas gráficas de navegación, etc.

- *Soporte a diversos usuarios.* Algunos ambientes son construidos para usuarios expertos, algunos para los usuarios novatos, y algunos tienen configuraciones que les permiten a los usuarios personalizar ambientes apropiados a su gusto. Es importante determinar si el ambiente puede soportar todos los tipos de usuarios.

- *Estilo de la presentación.* Estrechamente relacionado al tipo del usuario esta el estilo de la presentación. Algunos usuarios necesitan ver un detalle extenso, otros simplemente una abstracción. La presentación de información puede ser textual, gráfica, u otra.

- *Extensibilidad.* Es imposible anticipar todas las necesidades que una aplicación tendrá. Por consiguiente, es importante usar un ambiente que pueda adaptarse a lo largo de las necesidades de los usuarios y los proyectos.

2.5 Editores de ontologías

La creación de ontologías es una tarea desafiante. El desarrollo distribuido de ontologías necesita herramientas que permitan la sincronización entre los distintos agentes, es decir, entre las distintas aplicaciones que hacen uso o ayudan a definir la ontología. Pero adquirir una ontología de sólo un agente también es difícil, dado que tienen que hacerse explícitos conceptos que usualmente se encuentran implícitos, debido a hechos que se asumen, propiedades que siempre se cumplen, etc.

Un ejemplo de editor de ontologías es *Protégé*, el cual soporta ontologías y adquisición de conocimiento a un usuario novato. Es además, una *herramienta* que permite al usuario construir una ontología del dominio y personalizar formas de entrada de datos o puede verse también, como una *biblioteca* que otras aplicaciones pueden usar para acceder y desplegar bases de conocimiento. Permite la importación y exportación de ontologías *OWL*.

Otro ejemplo de editor de ontologías basado en la *Web* es *WebOnt*, que soporta la creación de ontologías sobre la *Web*. Existen además, editores de ontologías que soportan *RDF Schema* [28], siendo importante debido al alto grado de estandarización, aceptación y utilización que tiene dicho lenguaje.

A su vez *Oiled* es un editor de ontologías simples que le permite al usuario construir ontologías que usan *OIL* [29] y aunque con algunos problemas *DAML+OIL*. La intención detrás de *Oiled* es proporcionar un editor simple y gratuito que demuestra el uso y estimula el interés en *OIL*. *Oiled* no se piensa como un ambiente de desarrollo de ontología completo ya que no soporta el desarrollo de las ontologías de gran potencia, la migración e integración de las mismas, así como el versionado, aumentaciones y muchas otras actividades que están envueltas en la construcción de ontologías.

Un enfoque distinto, necesario para soportar el reuso de conocimiento formalmente representado entre los sistemas de *Inteligencia Artificial (IA*), donde es útil definir el vocabulario común en el que el conocimiento compartido se representa, es el de *Ontolingua*. Esta herramienta presenta un mecanismo para escribir ontologías en un formato canónico, tal que puedan traducirse fácilmente en una variedad de representaciones y razonadores de sistemas [22]. Esta cualidad permite mantener la ontología en una forma simple y legible para usarse en sistemas con diferentes sintaxis y capacidades de razonamiento.

On-To-Knowledge [30], emplea el poder de la aproximación ontológica para facilitar el manejo de conocimiento. El enfoque de su aplicación es el manejo de conocimiento en organizaciones grandes y distribuidas. A consecuencia se desarrolló dicha herramienta para tener un acceso inteligente a los grandes volúmenes semi-estructurados y ambientes basados en *Internet*, y de esta forma emplear el poder de las ontologías en el soporte del manejo de conocimiento de forma simple y eficaz, proporcionando y ayudando a mantener grandes cuerpos de texto y fuentes de información semi-estructuradas.

Otro editor, que soporta básicamente el lenguaje *DAML*, es *OnTo-Agents,* que fue pensado en el contexto del proyecto *OntoAgents* [31] para definir los términos que son posibles de usar para la información de marcado en páginas *Web*.

Los editores antes mencionados, son algunos de los más significativos, para ver una información más detallada sobre los editores de ontologías ver el Anexo 2 - Herramientas.

2.6 Elección de Lenguaje y herramienta

Luego de haber analizado y estudiado los distintos lenguajes y herramientas que permiten definir y utilizar las ontologías, se utilizará para llevar a cabo el proyecto, el lenguaje *OWL*, junto con la herramienta *Protégé*.

La elección de *OWL*, se debe al alto grado de aceptación y creciente uso que tiene dicho lenguaje en la actualidad, además de todas las posibilidades y facilidades que este lenguaje brinda. Esto sumado al hecho de que la elección de dicho lenguaje, deberá permitir una gran variedad de herramientas para gestionar la ontología, o lo que es lo mismo, que acote lo menos posible dicha selección.

Por otro lado, la elección de *Protégé*, está en gran medida ligada a la elección de *OWL* como lenguaje. Adicionalmente ayudó a su elección la facilidad de uso, su interfaz amigable y las utilidades que brinda.

A todo lo anteriormente mencionado se suma el hecho de que permite el uso de razonadores como *Racer*, permitiendo hacer consultas, definir vistas para cada tipo de instancias, etc. Al estar implementado en *Java*, es multiplataforma, permitiendo además el agregado de paquetes externos para su utilización (*Plugins*).

Capítulo 3 OntoChoike

3.1 Introducción

Una vez relevados y analizados los distintos lenguajes para definir ontologías y las herramientas existentes para gestionarlas, se realizará en este capitulo la descripción detallada del problema a ser resuelto.

Como se planteo en el capitulo anterior, la solución propuesta, que se presenta en la sección 3.3, utilizará *OWL* y *Protégé* como herramientas para resolver el problema de generación de descripciones en la *Web*.

3.2 Visión global del problema

El problema a resolver es la generación de una ontología representativa del sitio *Choike*. El sitio puede verse como un navegador de informes, el cual se encuentra dividido en distintas categorías (no disjuntas), organizadas en 2 niveles (Figura *3*).

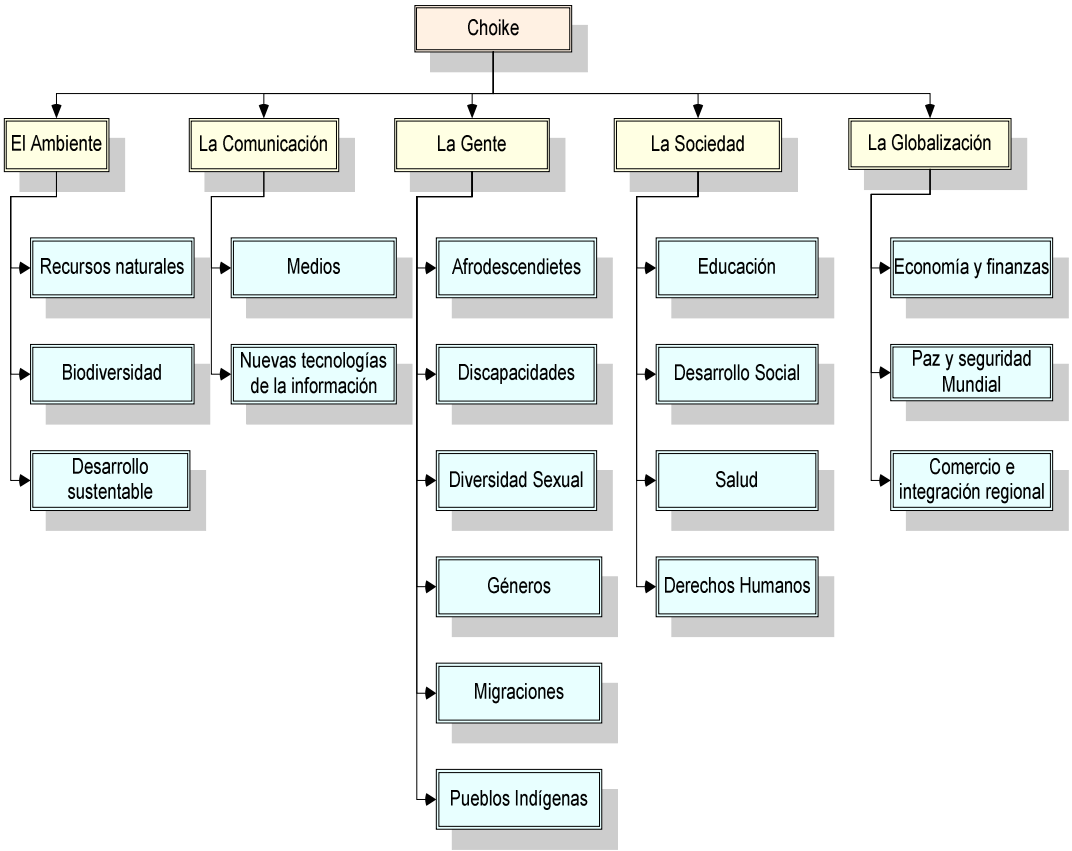

Figura 3 - Estructura de categorías de informes de Choike

Para dicha generación, se cuenta con la información contenida en las páginas (informes) e información brindada por *Choike* que posibilitan la clasificación de cada informe, en las distintas categorías a las que puede pertenecer.

Al disponer el sitio de los informes organizados en categorías, y considerando el hecho que una categorización, es ya en si misma una ontología, es razonable, pensar que dicha categorización debe formar parte de la ontología a ser generada.

Por otro lado, en los informes, aparece información diversa que sería útil quedara reflejada en la ontología, como ser distintas entidades nombradas, fechas, relaciones entre entidades, etc. Estas entidades, pueden a su vez, discriminarse según su tipo (organizaciones, personas, países, y demás). Existen diversas relaciones que resulta de interés poder reconocerlas, entre ellas el que una persona esté vinculada a una organización o disponga de un cargo, que una ciudad pertenezca a un país, o una organización esté relacionada con un país. En la Figura 4 puede observarse un esquema de lo que como mínimo debe formar parte de la ontología.

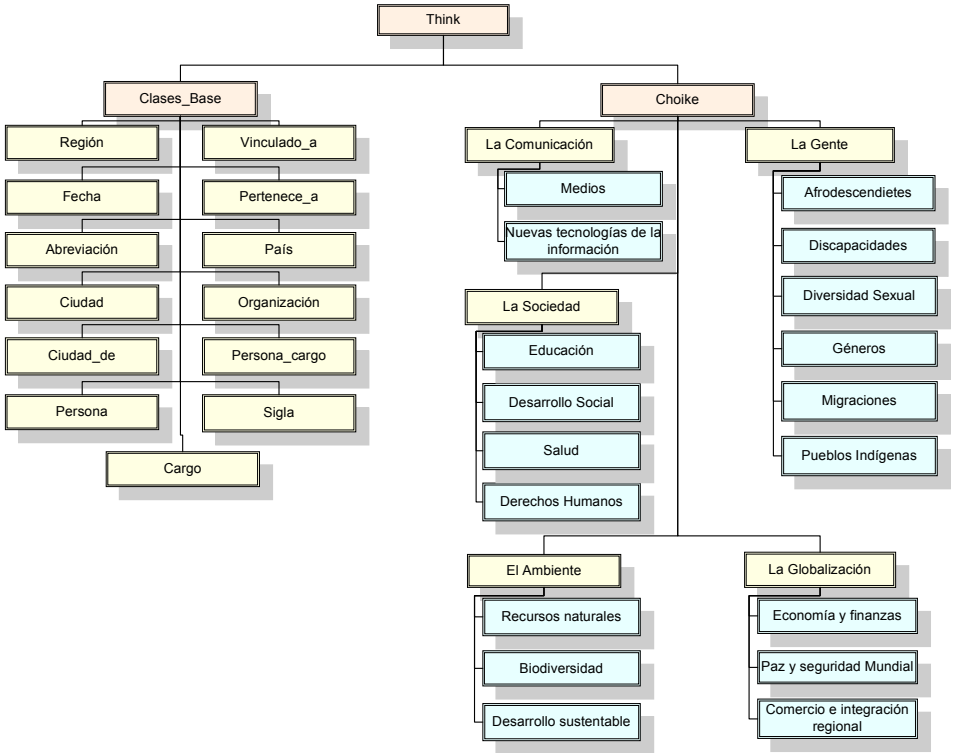

Figura 4 - Esquema de ontología

Resulta evidente, que cuanto más completa sea una ontología, mejores resultados se pueden obtener de ella. Por esto mismo, es que resulta lógico pensar que la ontología disponga desde su comienzo de información referente a distintas organizaciones, países y ciudades, aunque estas no se encuentren en los informes del sitio.

Otra forma de enriquecer la ontología es, por ejemplo, discriminar las organizaciones en documentos, eventos y organizaciones propiamente dichas. Es conveniente, además, poder saber que entidades y relaciones conforman un informe (a la vez que su número de ocurrencias), o a la inversa, dada una entidad o relación, saber en que informes se encuentran. Este tipo de información puede ser útil a la hora de realizar búsquedas de informes por similitud, o determinar él o los temas que trata un informe, etc.

Finalmente, la ontología a ser generada debe reflejar todo lo expuesto anteriormente, recibiendo el nombre de *ontología inicial* u *ontología base*, encontrándose en la Figura 5 un esquema de la misma.

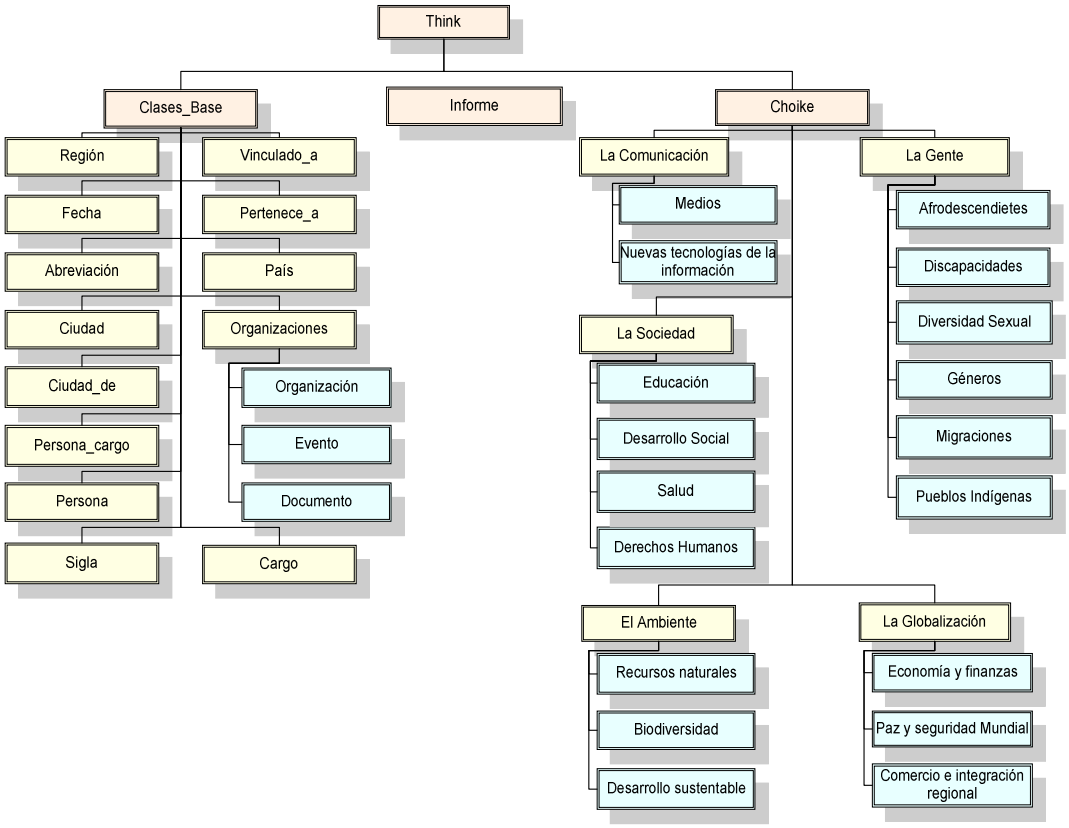

Figura 5 - Esquema de ontología base

3.3 Descripción general de la solución

La solución planteada para la generación de una ontología representativa del sitio *Choike* toma como base una ontología a la cual se le agregan datos adicionales, obtenidos mediante técnicas de extracción de información de las páginas del sitio (ver Figura 6).

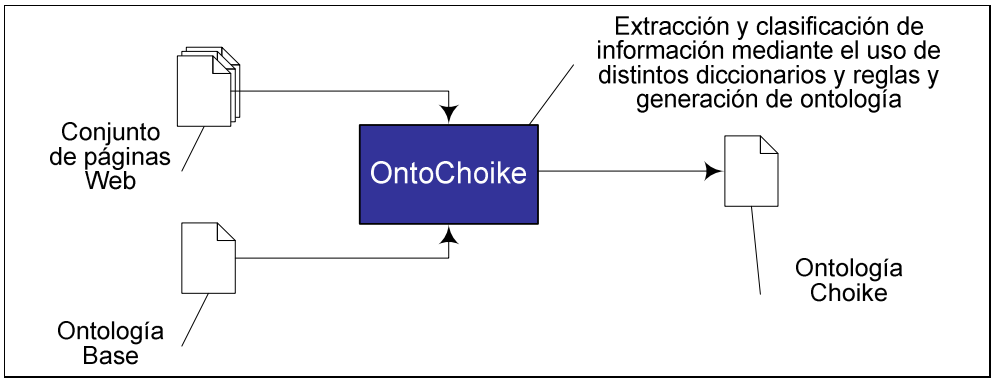

Figura 6 – Solución Propuesta

El problema original se divide en dos sub-problemas bien diferenciados. El primero, relativo a la extracción de información a partir de las páginas, y el segundo, encargado de actualizar la *ontología inicial* con los datos obtenidos.

3.3.1 Extracción de Información

Para cada informe (página), se obtiene el texto contenido en la misma, omitiendo partes de la página que no contienen información relevante, como ser el cabezal y pie de página. Se eliminan además, todas las marcas (*tags*) propias de las páginas *HTML* que contienen información referente a su presentación.

Seguidamente se comienza a extraer la información que ellos contienen. En principio podría pensarse en tener un gran diccionario que contenga los nombres de entidades a ser reconocidas, pero luego de un pequeño análisis se puede ver que esta solución no sería satisfactoria, dado que el conjunto de entidades a reconocer no es cerrado. Por ejemplo, los nombres de ciudades o países pueden considerarse como algo invariante (partiendo de la base que los países o ciudades no se crean o cambian de nombre en forma muy habitual) pero los nombres de personas u organizaciones distan mucho de serlo.

Por esta razón se hace necesario el reconocimiento también de nuevas entidades, o sea, de los cuales no se tiene información previa. Por lo que se hace necesaria la inferencia de información.

A partir de distintos diccionarios (nombres, países, ciudades, etc.) y reglas, se reconoce un conjunto abierto de nombres de entidades y relaciones. El cometido de los diccionarios es denotar si una palabra puede ser una posible fecha, o si pertenece al diccionario de nombres, países, etc. En cambio, las reglas permiten hacer distintas validaciones que posibilitan tener un mayor grado de exactitud en los datos reconocidos, filtrando por ejemplo fechas inválidas o ayudando a reconocer información adicional.

En esta etapa surgen problemas de ambigüedades semánticas, los cuales se presentan cuando la palabra que se está procesando pertenece a más de un diccionario. O sea, para los casos en los que una palabra pueda ser, por ejemplo, un posible nombre de persona o país. En estos casos se resuelven estas ambigüedades otorgando un orden de precedencia a los distintos tipos de instancias.

Para aclarar ideas, dada la palabra *Argentina*, la cual puede ser reconocida como el nombre de un país o persona, se le da preferencia al nombre del país. En consecuencia, en este contexto la palabra *Argentina* siempre denota el nombre de un país.

Luego, para cada posible dato relevante reconocido, se le agrega información referente a su potencial tipo de entidad, en qué posición fue encontrada, y un marco de palabras formada en lo general por las cinco palabras anteriores y las cinco próximas, dependiendo de a qué tipo de dato pertenezca. Ya que en el caso que el tipo candidato de la instancia, sea una organización, el marco se conforma con las siguientes quince palabras o cinco por ejemplo para el caso de que la instancia candidata sea una persona. Esto, se requiere para poder terminar de obtener el nombre de las entidades, dado que en este proceso sólo se devuelve la palabra clave encontrada, y hay que terminar de formar el nombre completo, en un proceso posterior. La decisión de cuantas palabras anteriores o siguientes son tomadas en cuenta, es resuelta empíricamente según el tipo candidato de instancia.

Adicionalmente, para los casos de ciudades y países, se agregan los distintos nombres a los que se puede estar haciendo referencia. Por ejemplo, al encontrarse la palabra "Reino", dado que esta es una palabra clave, asociada al nombre de país "Reino Unido", se le asigna el tipo de entidad candidato *País*, y además de agregarse las 5 palabras anteriores y 5 próximas, se agrega información referente a que "Reino Unido" es el posible nombre del país que se está reconociendo. Pudiendo en otros casos hacerse referencia a más de un país o ciudad.

Es importante saber también la posición dónde fue encontrada la palabra, para así evitar que se procese una palabra clave más de una vez.

Para afirmar ideas, dado el texto de entrada "Juan Pablo Pérez", se reconocen las palabras claves "Juan" y "Pablo", siendo clasificadas como potenciales nombre de personas (las que son representadas por medio de su nombre y apellido).

En este ejemplo, "Pablo", no debería de reconocerse como el nombre de otra persona, dado que claramente es parte del nombre de "Juan". He aquí que se hace relevante la posición en la que se encuentra cada palabra clave, pudiéndose así eliminar en un proceso posterior las palabras que fueron contempladas en los análisis de palabras claves previas.

Por otro lado, las organizaciones son discriminadas en tres categorías: documentos (Tratado, Carta, Acuerdo, Protocolo, etc.), eventos (Conferencia, Congreso, etc.) y organizaciones propiamente dichas (Organización, Universidad, Banco, etc.). Habiendo por consiguiente, un diccionario para cada una de estas categorías.

En el caso de las fechas se reconocen los formatos "MMMM del AAAA", "DD/MM/AAAA", "DD de MMMM de AAAA" y "PALABRA_CAPITAL'AA". Por este motivo, "26 de febrero", "12 de marzo de 1990" y URUGUAY'95 son algunos ejemplos de fechas válidas.

Es importante remarcar además, que para el caso del diccionario de ciudades y países, estos no sólo contienen la palabra clave que es usada para reconocerlas, sino que además contiene el nombre completo de las mismas. O sea, a partir del diccionario, se puede saber que *Los Ángeles* tiene como palabra clave a *Ángeles* y además pertenece a *Estados Unidos*, el cual a su vez tiene como palabra clave *Estados*.

Siguiendo con el proceso, se terminan de aplicar las reglas de reconocimiento, tanto de las entidades nombradas, como de las fechas detectadas.

De esta forma se finaliza la incorporación de toda la información encontrada para un cierto nombre de persona, fecha, país, etc.

Además se filtran los candidatos erróneos, como ser fechas inválidas, nombres de países inexistentes, etc., utilizándose para este fin distintas reglas y la información adicional incluida anteriormente.

Un ejemplo de regla de inferencia aplicada en esta instancia del proceso es la utilizada para reconocer nombres de personas, la cual consiste en asumir que un nombre de pila, seguido de letras capitales, y/o palabras comenzadas en mayúscula o artículos como ser “de” “del”, etc., son parte del nombre de una persona. De esta forma se puede reconocer que *Jorge del Campo* es el nombre de una persona, dado que *Jorge* pertenece al diccionario de nombres, *del* es un artículo de los contemplados en los nombres de personas y a su vez está seguido de una palabra comenzada en mayúscula.

El proceso que se realiza para reconocer nombres de organizaciones, eventos y documentos es extremadamente similar al de personas teniendo, además de la diferencia obvia de tomar las palabras claves de distintos diccionarios, el hecho de que los artículos que estos contemplan son diferentes, lo que se debe principalmente al diferente contexto de las mismas.

Continuando con el proceso de extracción y clasificación de información y a partir de las entidades antes reconocidas y filtradas, es que se reconocen distintos tipos de relaciones existentes entre ellas.

El problema principal que se encuentra en el reconocimiento de relaciones, viene dado por el hecho de que al análisis sintáctico del texto se le suma además uno semántico. Por lo que se buscan reglas que reflejen las relaciones y requirieran el menor análisis posible.

Este reconocimiento, es hecho de maneras variadas dependiendo principalmente del tipo de relación que se está reconociendo y dependen también en gran medida de como estas relaciones aparecen en los informes de *Choike*.

Igualmente, este problema resulta difícil de resolver, ya que para poder reconocer una relación existente entre dos entidades, primero necesitan ser reconocidas, lo que, como fue mencionado anteriormente, no es una tarea trivial. Si además a esto se le suma la riqueza y diversidad que posee el Idioma Español para permitir expresar la misma idea, o en este caso relaciones, el problema se dificulta aún más.

Además, no todas las relaciones que pueden existir entre dos entidades son relevantes en el dominio específico de *Choike*, estando a su vez ligadas en gran medida a los tipos de entidades que se reconocen. Por esto fueron acotados los distintos tipos de relaciones a ser reconocidas a las siguientes:

- *Abreviación*
  Asocia una organización, evento o documento con su correspondiente sigla o acrónimo.
  Ejemplo: "En el Fondo Monetario Internacional (*FMI*) se estudia la….."
  Relación: *Abreviación* (Fondo Monetario Internacional, *FMI*)

- *Vinculado_a*
  Vincula una persona a un documento, organización o evento.
  Ejemplo: "El presidente del Comité Pacificación, Juan Pérez, fue quien…."
  Relación: *Vinculado_a* (Juan Pérez, Comité Pacificación)

- *Pertenece_a*
  Vincula una organización, evento o documento a un país.
  Ejemplo: "Universidad de la República (Uruguay)…."
  Relación: *Pertenece_a* (Universidad de la República, Uruguay)

- *Persona_Cargo*
  Asocia una persona a un determinado cargo.
  Ejemplo1: "El presidente del Comité Pacificación, Juan Pérez, fue quien…."
  Relación: *Persona_Cargo* (Juan Pérez, presidente)
  Ejemplo2: "El ex-ministro de cultura Pedro Fernández…"
  Relación: *Persona_Cargo* (Pedro Fernández, ex-ministro)

- *Ciudad_de*
  Vincula una ciudad con el país al que pertenece.
  Ejemplo: "…la cual fue dictada en Kyoto, Japón."
  Relación: *Ciudad_de* (Kyoto, Japón)

El método utilizado para reconocer los distintos tipos de relaciones, se realiza haciendo uso de diversas reglas. Por ejemplo, la relación *Vinculado_a* es reconocida cuando se encuentra una instancia de una organización, evento o documento y una instancia de una persona a una distancia menor a 20 palabras.

Como ejemplo, dado el texto de la Figura 7, se reconocen del mismo la persona *Carlos Fernández* y la organización *Universidad de Montevideo*, y como la distancia entre las mismas es menor a 20, se reconoce además la relación *Vinculado_a*, entre las mismas.

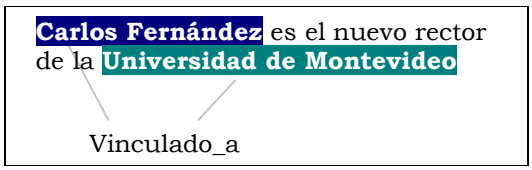

Figura 7 - Ejemplo Relación Vinculado_a

En otros casos, como ser en la relación *Abreviación*, se apuesta fuertemente a las reglas que, por ejemplo, reconocen esta relación cuando encuentran el nombre de una organización, evento o documento, seguido de una sigla entre paréntesis, o seguido de un guión (-) y una sigla, o los casos inversos. Un ejemplo de esto se da en la Figura 8.

Figura 8 - Ejemplo Relación Abreviación

Por último y luego de realizar todo el proceso antes mencionado, se obtienen las distintas entidades reconocidas y clasificadas de las distintas páginas, junto con las relaciones que entre ellas se observan. En la Figura 9 se puede apreciar los distintos tipos de entidades y relaciones que se contemplan.

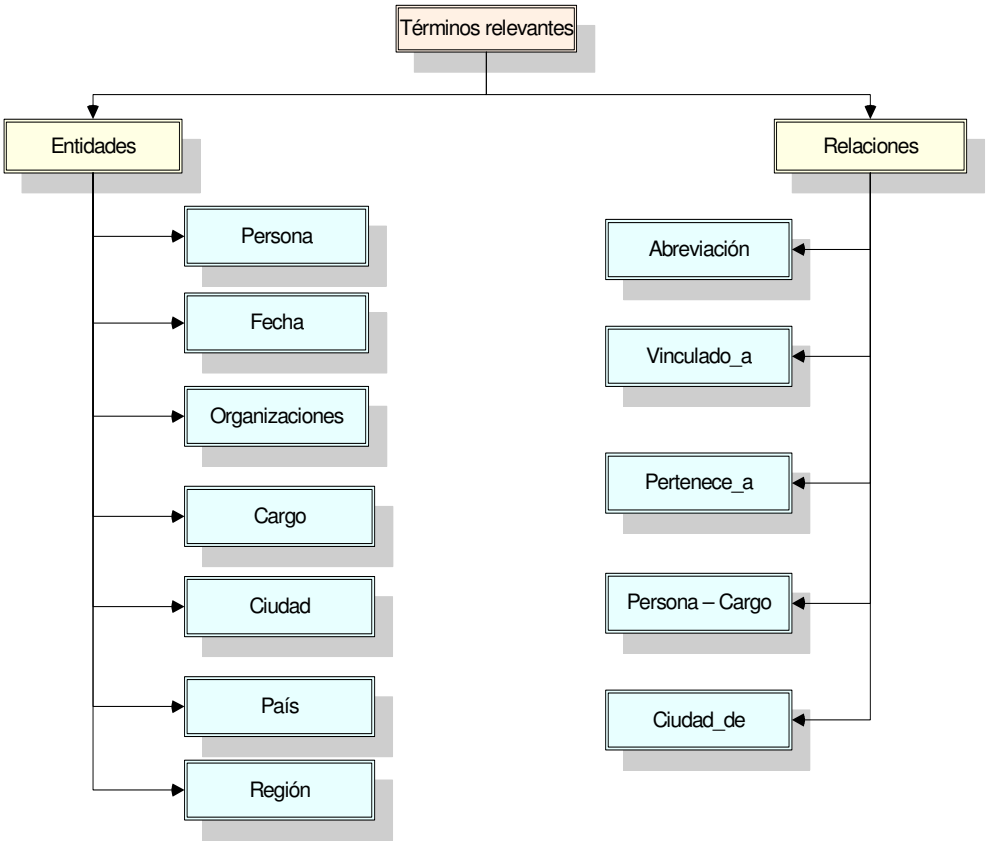

Figura 9 - Entidades y relaciones

3.3.2 Generación de la ontología

Luego de haber reconocido y clasificado la información contenida en los distintos informes de Choike, se pasa a la generación de la ontología. Lo cual consiste en ir actualizando o agregando dicha información en la *ontología inicial*.

Para esto, para cada entidad reconocida se verifica si ya pertenece a la ontología y en caso negativo se le agrega. En todo caso se agrega una referencia a dicha instancia en la categoría del informe a la cual ella pertenece, y una referencia de la instancia al informe. Adicionalmente se actualiza la cantidad de ocurrencias de la instancia en el informe. Cabe destacar que la forma en que se obtiene la categoría a la cual pertenece un informe es brindada por *Choike*.

Para las relaciones, se realiza el mismo tratamiento que para las entidades, con el agregado que además de corroborarse si las entidades involucradas en dicha relación (siempre binarias), pertenecen a la ontología, se agregan las referencias a dichas instancias.

La idea de tener una ontología inicial a la cual se le agrega información para luego obtener otra ontología, permite que esta solución sea incremental, o sea, que la ontología generada en cierto momento, pueda ser la ontología inicial en un momento posterior.

3.4 Implementación

La ontología inicial fue editada con *Protégé* (archivo OWL) a la cual se le agregan datos adicionales reconocidos y clasificados de las páginas del sitio. Como resultado del proceso se obtiene un archivo *OWL* conteniendo la ontología generada.

La división del problema original en dos sub-problemas da lugar a las dos etapas en las que se divide el funcionamiento del motor que conforma la solución general.

Estas etapas, apreciadas en la Figura 10, son:

- **text2xml**, procesa las páginas (informes), y extrae la información reconocida, generando un archivo *XML*.
- **xml2owl**, se encarga de procesar las entidades reconocidas en la etapa anterior y generar la ontología en formato *OWL*.

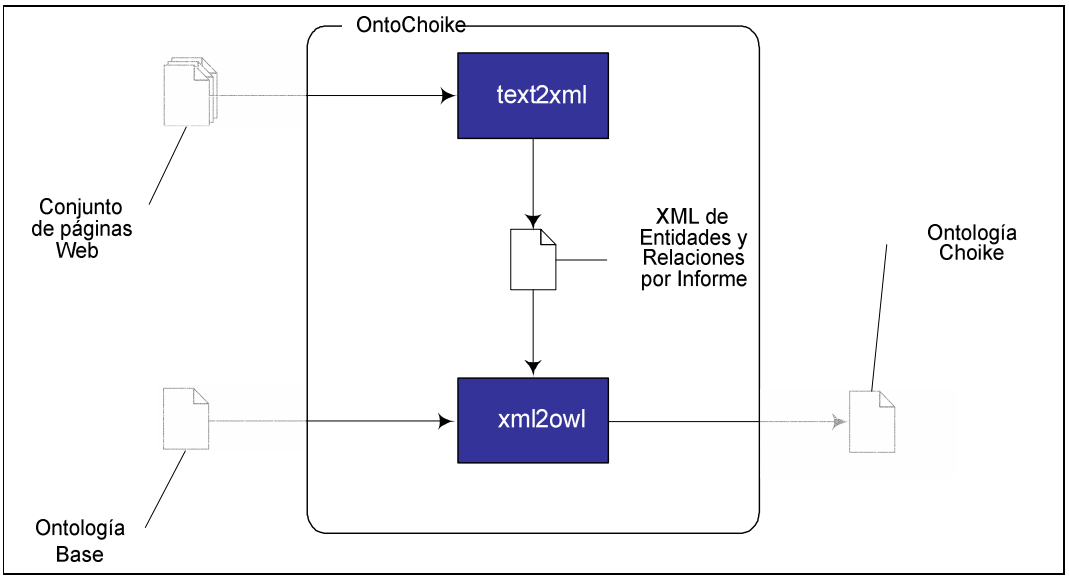

Figura 10 - Arquitectura OntoChoike

A continuación, se describen cada una de las etapas mencionadas.

3.4.1 text2xml

La idea principal de esta parte es la extracción de información. A partir de un conjunto de páginas *Web* dadas y ciertas reglas, se reconoce información que podría ser relevante para su posterior procesamiento y generación de una ontología.

La etapa se encuentra dividida en los procesos de extracción del texto, *tokenización*, reconocimiento de términos relevantes, filtrado y por último, reconocimiento de relaciones. Un esquema de su arquitectura se puede apreciar en la Figura 11.

Esta etapa fue escrita prácticamente en su totalidad en *Java*, dado el alto grado de portabilidad que con este se obtiene y la disponibilidad de diversos paquetes que permiten mejorar la implementación de la solución, brindando además soporte para la gestión de archivos *XML*, *XSLT* o *HTML*.

El único módulo que fue implementado en otro lenguaje fue el de reconocimiento de términos relevantes, que fue realizado en el lenguaje *XSLT*.

Esta elección se debió a que el mismo soporta en forma nativa el uso de expresiones regulares, y fue concebido para ser usado conjuntamente con archivos *XML*, simplificando en gran medida la implementación de la solución.

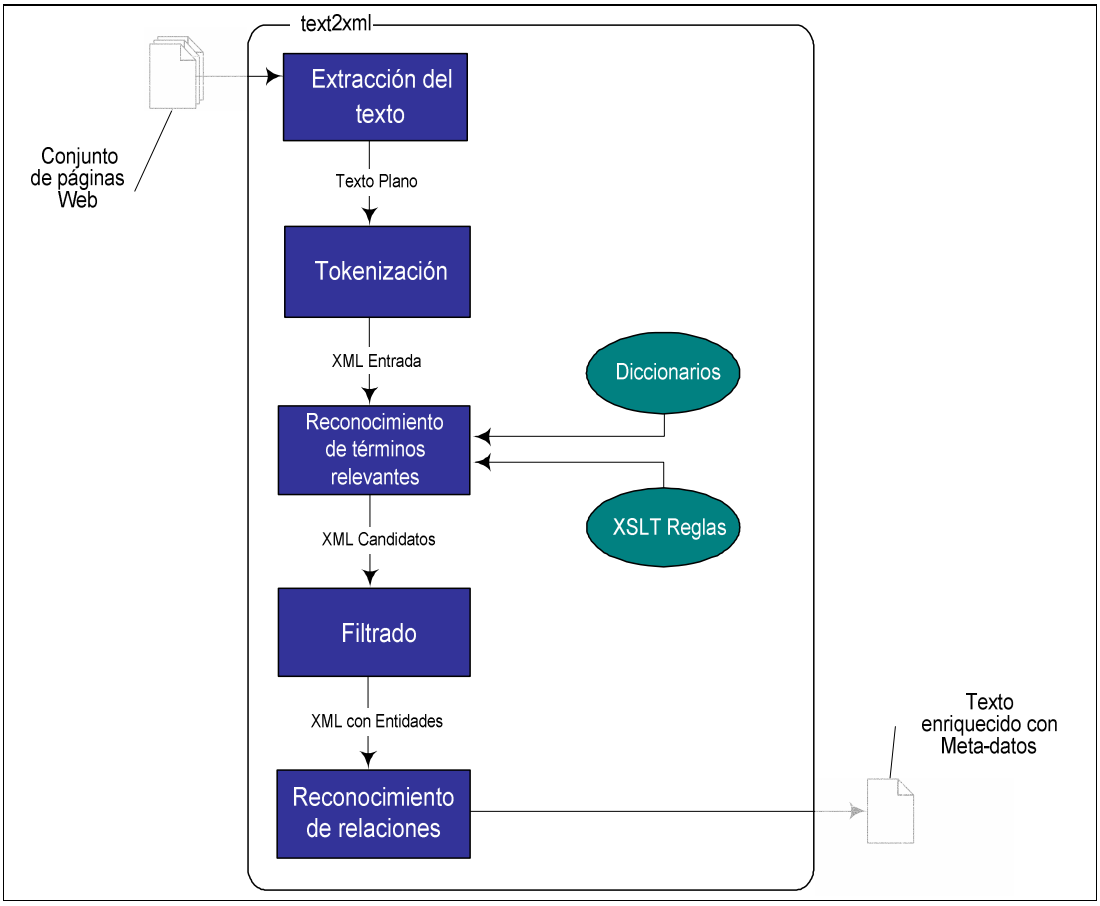

Figura 11 - Arquitectura text2xml

3.4.1.1 Extracción de texto

La finalidad de este proceso, es la obtención del texto contenido en una página *Web*, omitiendo partes de la página que no contienen información relevante, como ser el cabezal y pie de página. Un ejemplo de esto se encuentra en la Figura 12.

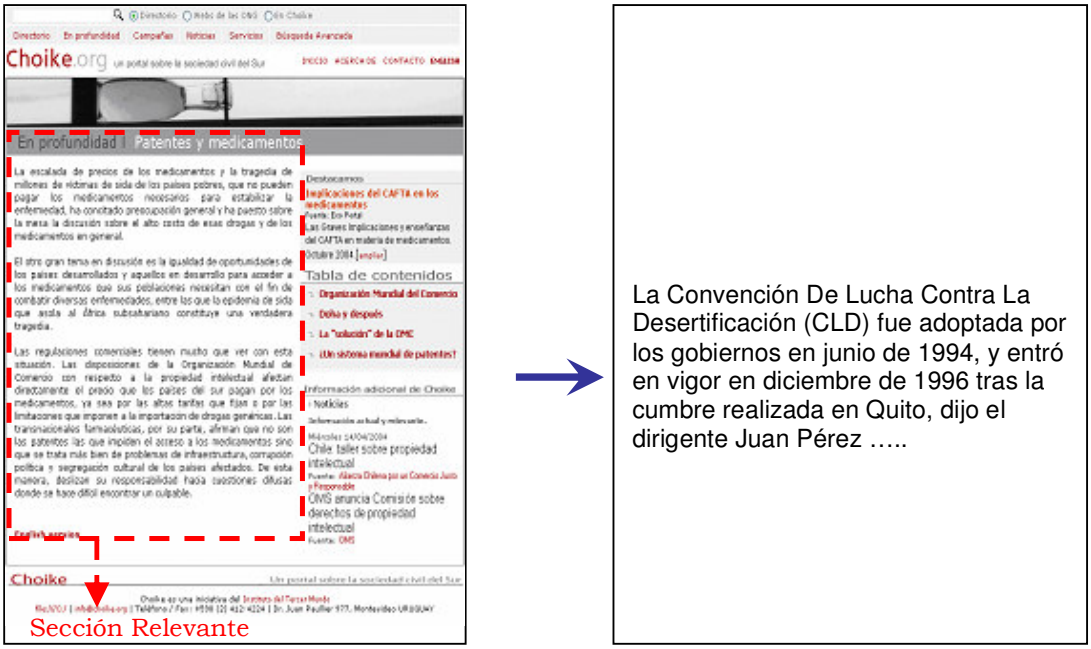

Figura 12 - Ejemplo de Extracción de Texto

Para hacer posible la extracción del texto contenido en las páginas *Web*, se hizo uso del paquete de *Java HTMLparser* [35] al cual se le realizaron pequeñas modificaciones con el objetivo de amoldarlo lo más posible a las necesidades del caso.

3.4.1.2 Tokenización

El propósito de este proceso es la generación de un documento *XML* a partir del texto plano de entrada, el cual contendrá la misma información, pero con una estructura de *XML* como la apreciada en la Figura 13.

Esto es necesario para su posterior análisis con *XSLT* y poder así seguir con las demás etapas de su procesamiento.

El manejo de documentos *XML*, se realizó mediante el paquete de *Java JDOM* [36], el cual brinda soporte a todas las funciones que posibilitan su navegación, permitiendo recorrer los nodos que cumplen con ciertas condiciones, obtener el valor de los mismos, etc.

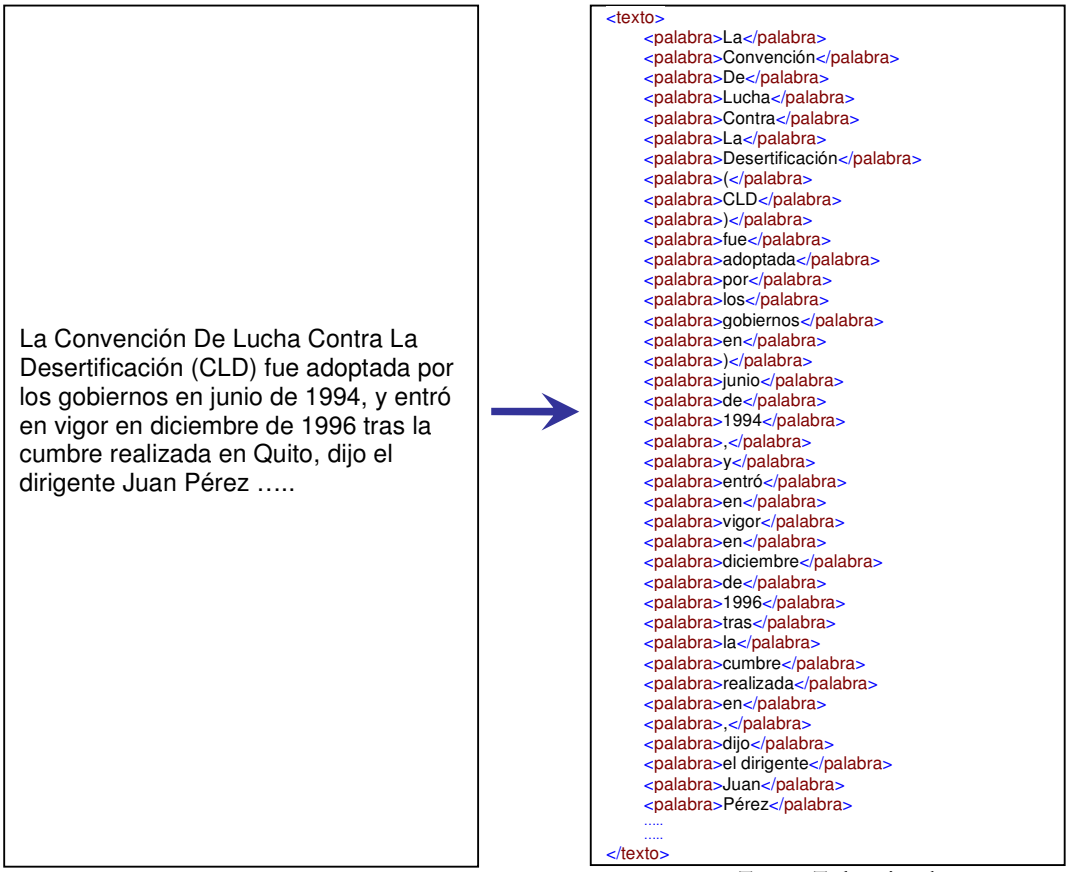

Figura 13 - Ejemplo de Tokenización

3.4.1.3 Reconocimiento de términos relevantes

Cabe destacar que aquí es donde comienza el proceso de inferencias y reconocimiento de información.

Este proceso, aunque no agrega demasiada lógica, ni aplica procesos de inferencia avanzados, es lo suficientemente eficaz como para descartar los datos que no aportan información, dejando para su posterior procesamiento solo aquellos datos de los cuales se puede obtener algún tipo de información relevante.

Para poder clasificar un dato como información relevante, basta con que la palabra clave pertenezca a algunos de los diccionarios considerados por la herramienta, o cumpla con ciertas expresiones regulares, que denoten, por ejemplo, el ser candidatas a fechas, etc.

La aplicación reconoce distintos tipos de entidades nombradas, al igual que fechas y variados tipos de relaciones que se pueden encontrar entre las entidades nombradas.

Los distintos diccionarios utilizados (almacenados en archivos *XML*) así como también sus estructuras, son presentados en la
Figura 14.

Diccionario	Estructura
cargos.xml	<cargos> … <cargo value="presidente"/> … </cargos>
organizaciones.xml	<organizaciones> … <organización value="Banco"/> … </organizaciones>
documentos.xml	<documentos> … <documento value="Acuerdo"/> … </documentos >
eventos.xml	<eventos> … <evento value="Conferencia"/> … </eventos>
fechas.xml	<fechas> … <mes value="enero"/> … </fechas>
nombres.xml	<nombres> … <nombre value="Abel"/> … /nombres>
paises.xml	<paises> … <pais value="Alemania"> … <ciudad value="Achen"/> … </pais> … <pais value="Antillas" nombre="Antillas holandesas"> … <ciudad value="Saint" nombre="Saint Eustatius"/> … </pais> … </paises >
regiones.xml	<regiones> … <region value="Caribe"/> … <region value="Oriente" nombre="Medio Oriente"/> … </regiones>

Figura 14 - Estructura de diccionarios

Las expresiones regulares (reglas) son aplicadas por medio de un archivo *XSLT* que se encarga de utilizar los diccionarios antes mencionados conjuntamente con el documento *XML* obtenido en la etapa de extracción del texto. Utilizando *JDOM*, también para el uso de archivo *XSLT*.

En la Figura 15, puede observarse un ejemplo de las entidades candidatas reconocidas en esta etapa.

```
<datos>
 <dato tipo="nombreEvento">
  <valor>Convención</valor><posicion>2</posicion>
  <anteriores><palabra>La</palabra></anteriores>
  <siguientes>
   <palabra>De</palabra><palabra>Lucha</palabra><palabra>Contra</palabra><palabra>La</palabra>
   <palabra>Desertificación</palabra><palabra>(</palabra><palabra>CLD</palabra><palabra>)</palabra>
   <palabra>fue</palabra><palabra>adoptada</palabra>  <palabra>por</palabra><palabra>los</palabra>
   <palabra>gobiernos</palabra><palabra>en</palabra>  <palabra>junio</palabra>
  </siguientes>
 </dato>
 <dato tipo="sigla">
  <valor>CLD</valor><posicion>9</posicion>
  <anteriores>
   <palabra>Lucha</palabra><palabra>Contra</palabra><palabra>La</palabra><palabra>Desertificación</palabra><palabra>(</palabra>
  </anteriores>
  <siguientes>
   <palabra>)</palabra><palabra>fue</palabra>  <palabra>adoptada</palabra><palabra>por</palabra><palabra>los</palabra>
   <palabra>gobiernos</palabra><palabra>en</palabra>  <palabra>junio</palabra><palabra>de</palabra><palabra>1994</palabra>
   <palabra>,</palabra><palabra>y</palabra><palabra>entró</palabra><palabra>en</palabra><palabra>vigor</palabra>
  </siguientes>
 </dato>
 <dato tipo="fecha">
  <valor>junio</valor>  <posicion>17</posicion>
  <anteriores>
   <palabra>adoptada</palabra><palabra>por</palabra><palabra>los</palabra><palabra>gobiernos</palabra><palabra>en</palabra>
  </anteriores>
  <siguientes>
   <palabra>de</palabra><palabra>1994</palabra><palabra>,</palabra><palabra>y</palabra><palabra>entró</palabra>
   <palabra>en</palabra><palabra>vigor</palabra><palabra>en</palabra><palabra>diciembre</palabra><palabra>de</palabra>
   <palabra>1996</palabra><palabra>tras</palabra><palabra>la</palabra><palabra>cumbre</palabra><palabra>realizada</palabra>
  </siguientes>
 </dato>
 <dato tipo="fecha">
  <valor>1994</valor><posicion>19</posicion>
  <anteriores>
   <palabra>los</palabra><palabra>gobiernos</palabra><palabra>en</palabra><palabra>junio</palabra><palabra>de</palabra>
  </anteriores>
  <siguientes>
   <palabra>,</palabra><palabra>y</palabra><palabra>entró</palabra><palabra>en</palabra><palabra>vigor</palabra>
   <palabra>en</palabra><palabra>diciembre</palabra><palabra>de</palabra><palabra>1996</palabra><palabra>tras</palabra>
   <palabra>la</palabra><palabra>cumbre</palabra><palabra>realizada</palabra><palabra>en</palabra><palabra>Quito</palabra>
  </siguientes>
 </dato>
 <dato tipo="fecha">
  <valor>diciembre</valor><posicion>26</posicion>
  <anteriores>
   <palabra>y</palabra><palabra>entró</palabra><palabra>en</palabra><palabra>vigor</palabra><palabra>en</palabra>
  </anteriores>
  <siguientes>
   <palabra>de</palabra><palabra>1996</palabra><palabra>tras</palabra><palabra>la</palabra><palabra>cumbre</palabra>
   <palabra>realizada</palabra><palabra>en</palabra><palabra>Quito</palabra><palabra>,</palabra><palabra>dijo</palabra>
   <palabra>el</palabra><palabra>dirigente</palabra><palabra>Juan</palabra><palabra>Pérez</palabra><palabra>?</palabra>
  </siguientes>
 </dato>
 <dato tipo="fecha">
  <valor>1996</valor><posicion>28</posicion>
  <anteriores>
   <palabra>en</palabra><palabra>vigor</palabra><palabra>en</palabra><palabra>diciembre</palabra><palabra>de</palabra>
  </anteriores>
  <siguientes>
   <palabra>tras</palabra><palabra>la</palabra><palabra>cumbre</palabra><palabra>realizada</palabra><palabra>en</palabra>
   <palabra>Quito</palabra><palabra>,</palabra><palabra>dijo</palabra><palabra>el</palabra><palabra>dirigente</palabra>
   <palabra>Juan</palabra><palabra>Pérez</palabra><palabra>?</palabra><palabra>.</palabra>  <palabra>.</palabra>
  </siguientes>
 </dato>
 <dato tipo="nombreCiudad">
  <valor>Quito</valor>
  <nombre/><valorPais>Ecuador</valorPais><posicion>34</posicion>
  <anteriores>
   <palabra>tras</palabra><palabra>la</palabra><palabra>cumbre</palabra><palabra>realizada</palabra><palabra>en</palabra>
  </anteriores>
  <siguientes>
   <palabra>,</palabra><palabra>dijo</palabra><palabra>el</palabra><palabra>dirigente</palabra><palabra>Juan</palabra>
   <palabra>Pérez</palabra><palabra>?</palabra><palabra>.</palabra><palabra>.</palabra>
  </siguientes>
 </dato>
 <dato tipo="nombrePersona">
  <valor>Juan</valor><posicion>39</posicion>
  <anteriores>
   <palabra>Quito</palabra><palabra>,</palabra>  <palabra>dijo</palabra><palabra>el</palabra><palabra>dirigente</palabra>
  </anteriores>
  <siguientes>
   <palabra>Pérez</palabra><palabra>?</palabra><palabra>.</palabra><palabra>.</palabra>
  </siguientes>
 </dato>
</datos>
```

Figura 15 – Ejemplo de reconocimiento de términos

3.4.1.4 Filtrado

Es en esta etapa donde finalmente al documento *XML* obtenido en la etapa anterior se le aplican nuevamente expresiones regulares, esta vez en *Java*, para terminar de filtrar y reconocer entidades y relaciones.

De esta forma, se finaliza la incorporación de toda la información encontrada para un cierto nombre de persona, fecha, etc. Se filtran además los candidatos erróneos, como ser fechas inválidas, nombres de países inexistentes, etc., utilizándose para este fin, expresiones regulares y la información adicional incluida en la etapa anterior.

En el *Anexo 3 – Expresiones Regulares* se encuentra información más detallada, sobre las reglas de inferencia aplicadas para reconocer nombre de organizaciones, personas, etc.

Un ejemplo de regla de inferencia aplicada en esta instancia del proceso es la utilizada para reconocer nombres de personas, la cual consiste en asumir que un nombre de pila, seguido de letras capitales, y/o palabras comenzadas en mayúscula o artículos como ser “de” “del”, etc., son parte del nombre de una persona. En la Figura 16 se puede apreciar un ejemplo del mismo.

```
<dato tipo="nombrePersona">
    <valor>Juan</valor>
    <posicion>4</posicion>
    <anteriores>
        <palabra>El</palabra>
        <palabra>asesor</palabra>
        <palabra>de</palabra>
        <palabra>la</palabra>
        <palabra>presidencia</palabra>
    </anteriores>
    <siguientes>
        <palabra>Pablo</palabra>
        <palabra>da</palabra>
        <palabra>Silva</palabra>
        <palabra>López</palabra>
        <palabra>comunicó</palabra>
    </siguientes>
</dato>
```

Dato devuelto en la etapa de reconocimiento de términos relevantes

```
<entidad tipo="persona">
    <valor>Juan Pablo da Silva López</valor>
    <posición>4</posición>
</entidad>
```

Dato terminado de procesar

Figura 16 – Ejemplo de nombre de persona

En la Figura 17, puede observarse un ejemplo de la salida de esta etapa.

```
<entidades>
    <entidad tipo="evento">
        <valor>Convención De Lucha Contra La Desertificación</valor>
        <posición>2</posición>
    </entidad>
    <entidad tipo="sigla">
        <valor>CLD</valor>
        <posición>9</posición>
    </entidad>
    <entidad tipo="fecha">
        <valor>junio de 1994</valor>
        <posición>17</posición>
    </entidad>
    <entidad tipo="fecha">
        <valor>diciembre de 1996</valor>
        <posición>26</posición>
    </entidad>
    <entidad tipo="ciudad">
        <valor>Quito</valor>
        <posición>34</posición>
    </entidad>
    <entidad tipo="persona">
        <valor>Juan Pérez</valor>
        <posición>39</posición>
    </entidad>
</entidades>
```

Figura 17 – Ejemplo de salida de la etapa de filtrado

3.4.1.5 Reconocimiento de relaciones

En esta etapa, y a partir de las entidades antes reconocidas y filtradas, se reconocen distintos tipos de relaciones existentes entre ellas.

En la
Figura 18, puede verse un ejemplo del reconocimiento de relaciones.

```
<relaciones>
    <relación tipo="abreviación">
        <evento>Convención De Lucha Contra La Desertificación</evento>
        <sigla>CLD</sigla>
    </relación>
    <relación tipo="ciudad_de">
        <ciudad>Quito</ciudad>
        <país>Ecuador</país>
    </relación>
</relaciones>
```

Figura 18 – Ejemplo de salida del reconocimiento de relaciones

Una descripción más detallada sobre la forma empleada para reconocer relaciones, se encuentra en el *Anexo 3 – Expresiones Regulares*.

3.4.1.6 Salida

La salida de este proceso, es un archivo *XML* (Figura 19), donde se presentan los datos encontrados por página (informe), de qué tipo son y las relaciones que pueden tener con otras entidades.

```
<reconocimiento>
    <pagina archivo="ejemplo.html">
        <entidades>
            <entidad tipo="evento">
                <valor>Convención De Lucha Contra La Desertificación</valor>
                <posición>2</posición>
            </entidad>
            <entidad tipo="sigla">
                <valor>CLD</valor>
                <posición>9</posición>
            </entidad>
            <entidad tipo="fecha">
                <valor>junio de 1994</valor>
                <posición>17</posición>
            </entidad>
            <entidad tipo="fecha">
                <valor>diciembre de 1996</valor>
                <posición>26</posición>
            </entidad>
            <entidad tipo="ciudad">
                <valor>Quito</valor>
                <posición>34</posición>
            </entidad>
            <entidad tipo="persona">
                <valor>Juan Pérez</valor>
                <posición>39</posición>
            </entidad>
        </entidades>
        <relaciones>
            <relación tipo="abreviación">
                <evento>Convención De Lucha Contra La Desertificación</evento>
                <sigla>CLD</sigla>
            </relación>
            <relación tipo="ciudad_de">
                <ciudad>Quito</ciudad>
                <país>Ecuador</país>
            </relación>
        </relaciones>
    </pagina>
</reconocimiento>
```

Figura 19 - XML Salida de text2xml

3.4.2 xml2owl

Esta etapa tiene como principal objetivo la generación de una ontología en formato *OWL*. Esto (como se puede apreciar en la Figura 20) a partir de un archivo *XML* de entrada (salida de *text2xml*) y una ontología base (archivo *OWL*).

La etapa, se encuentra dividida en dos procesos. El primero de ellos se encarga de la realización del procesamiento de entidades y el segundo efectúa posteriormente el procesamiento de las relaciones encontradas.

Esta etapa en su conjunto fue realizada en *Java*, usándose para manipular la ontología base e ir generando la ontología final el paquete *JDOM*. Este paquete también se utilizó para acceder al catálogo de informes y al archivo que contiene la salida de text2xml.

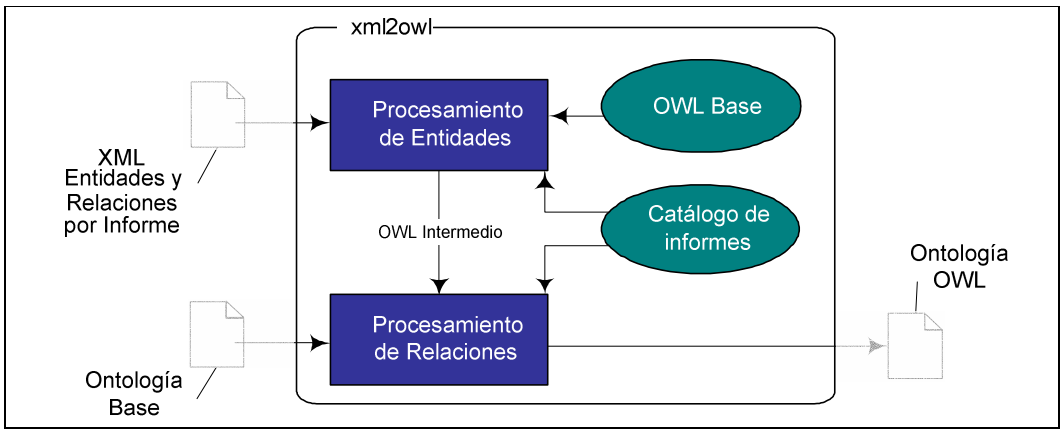

Figura 20 - Arquitectura xml2owl

El procesamiento de entidades y relaciones se realiza agregando la información que corresponda al *OWL Base*, a partir del archivo *XML* generado por *text2xml*. Para esto es necesario además saber la categoría a la cuál pertenece un informe, obteniéndose la misma de un catálogo proporcionado por *Choike*.

La salida de este proceso, es un documento *OWL* (Figura 21) conteniendo la ontología generada.

```
……
<Informe rdf:ID="Informe_141">
        <Código rdf:datatype="http://www.w3.org/2001/XMLSchema#string">141</Código>
        <Cantidad_Entidades rdf:datatype="http://www.w3.org/2001/XMLSchema#string">6</Cantidad_Entidades>
        <Cantidad_Relaciones
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">2</Cantidad_Relaciones>
    </Informe>
    <Economía_y_Finanzas rdf:ID="Economía_y_Finanzas_Instancias">
        <Instancias rdf:resource="#evento_2692"/>
        <Contiene_Informes rdf:resource="#Informe_141"/>
        <Instancias rdf:resource="#sigla_2693"/>
        <Instancias rdf:resource="#fecha_2694"/>
        <Instancias rdf:resource="#fecha_2695"/>
        <Instancias rdf:resource="#ciudad_856"/>
        <Instancias rdf:resource="#persona_2696"/>
        <Instancias rdf:resource="#abreviación_2697"/>
        <Instancias rdf:resource="#ciudad_de_2698"/>
    </Economía_y_Finanzas>
    <evento rdf:ID="evento_2692">
        <Valor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Convención De Lucha Contra La
Desertificación</Valor>
        <En_Informe rdf:resource="#Informe_141" rdf:ocurrencias="2"/>
    </evento>
    <sigla rdf:ID="sigla_2693">
        <Valor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">CLD</Valor>
        <En_Informe rdf:resource="#Informe_141" rdf:ocurrencias="2"/>
    </sigla>
    <fecha rdf:ID="fecha_2694">
        <Valor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">junio de 1994</Valor>
        <En_Informe rdf:resource="#Informe_141" rdf:ocurrencias="2"/>
    </fecha>
    <fecha rdf:ID="fecha_2695">
        <Valor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">diciembre de 1996</Valor>
        <En_Informe rdf:resource="#Informe_141" rdf:ocurrencias="2"/>
    </fecha>
    <persona rdf:ID="persona_2696">
        <Valor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Juan Pérez</Valor>
        <En_Informe rdf:resource="#Informe_141" rdf:ocurrencias="2"/>
    </persona>
    <abreviación rdf:ID="abreviación_2697">
        <De rdf:resource="#evento_2692"/>
        <A rdf:resource="#sigla_2693"/>
        <En_Informe rdf:resource="#Informe_141" rdf:ocurrencias="2"/>
    </abreviación>
    <país rdf:ID="país_2699">
        <Valor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Ecuador</Valor>
    </país>
    <ciudad_de rdf:ID="ciudad_de_2698">
        <De rdf:resource="#ciudad_856"/>
        <A rdf:resource="#país_2699"/>
        <En_Informe rdf:resource="#Informe_141" rdf:ocurrencias="2"/>
    </ciudad_de>
    <La_globalización rdf:ID="La_globalización_Instancias">
        <Instancias rdf:resource="#evento_2692"/>
        <Contiene_Informes rdf:resource="#Informe_141"/>
        <Instancias rdf:resource="#sigla_2693"/>
        <Instancias rdf:resource="#fecha_2694"/>
        <Instancias rdf:resource="#fecha_2695"/>
        <Instancias rdf:resource="#ciudad_856"/>
        <Instancias rdf:resource="#persona_2696"/>
        <Instancias rdf:resource="#abreviación_2697"/>
        <Instancias rdf:resource="#ciudad_de_2698"/>
    </La_globalización>
…..
```

Figura 21 - OWL Salida de xml2owl

Capítulo 4 Resultados

4.1 Introducción

A continuación, se presentan los resultados obtenidos con la solución *OntoChoike*. Cabe aclarar que todas las pruebas, fueron realizadas en una computadora de escritorio equipada con un procesador AMD XP 2000+, 640 MB de RAM, corriendo el sistema operativo Windows XP Professional SP2 y Java Virtual Machine 1.4.2.

4.2 OntoChoike

Los resultados obtenidos para *OntoChoike*, fueron analizados según las dos etapas que lo conforman. Para el caso de *xml2owl*, lo importante es el volumen de los datos que contiene la ontología generada, mientras que en lo referente a *text2xml*, si bien es importante dicha medida, es indiscutiblemente más importante analizar la calidad y correctitud de los mismos, dado que de obtenerse un tamaño considerable de datos, pero incorrectos, no sólo no ayuda a generar una ontología específica, sino que además puede degenerar la ontología en una incorrecta.

Por lo anterior, es que para este caso, se analizaron los resultados con el fin de obtener medidas de correctitud y calidad de los mismos.

4.2.1 text2xml

4.2.1.1 Forma de evaluación

Para poder realizar la estimación de los errores cometidos al reconocer información, es necesario reconocer y clasificar la información relevante en forma manual. Por tal motivo se tomaron en forma aleatoria, veinte informes de *Choike* para ser usados como corpus lingüístico, con la consiguiente clasificación manual de sus datos relevantes.

Estos informes fueron divididos en dos grupos de diez informes cada uno. Los primeros diez, se utilizaron como datos de entrenamiento, a partir de los diccionarios y reglas que fueron definidos en función del contenido y estructura globales de los informes del sitio, se les hicieron las modificaciones necesarias con el objetivo de reconocer la mayor cantidad de información relevante, cometiendo a su vez el menor error posible. Los restantes diez informes, fueron utilizados para la realización de la evaluación.

Para estimar los errores cometidos en el reconocimiento, se usaron las medidas de *recuperación* (que determina la cantidad de datos reconocidos) y *precisión* (que mide la calidad de los datos reconocidos), siendo sus fórmulas las siguientes:

$$\textit{recuperación} = VP / (VP + FN)$$

$$\textit{precisión} = VP / (VP + FP)$$

Donde

VP = Instancias reconocidas correctamente por la aplicación
FN = Instancias no reconocidas, pero que debían haberse reconocido
FP = Instancias reconocidas erróneamente por la aplicación

Adicionalmente, se usó la medida de *F-measure* (con $\alpha = 0{,}50$) que combina las dos medidas anteriores, siendo su fórmula la siguiente:

$$F\text{-}measure\ (\alpha) = (\alpha \,.\, precisión^{-1} + (1-\alpha) \,.\, recuperación^{-1})^{-1}$$

4.2.1.2 Resultados obtenidos

En la Figura 22, se puede observar la información referente a los diccionarios utilizados, como ser la cantidad de ciudades, países, nombres de personas, palabras claves para reconocer organizaciones, etc.

Diccionarios	Cantidad
organizaciones	40
documentos	16
eventos	15
países	205
ciudades	1372
nombres	711
fechas	12
regiones	11
cargos	45

Figura 22 - Información de los diccionarios

Como se mencionó en la sección anterior, un subconjunto del corpus fue utilizado para estimar la *precisión*, *recuperación* y *F-measure* de la etapa Para estos diez informes, se insumió un tiempo total de procesamiento de 2 minutos 14 segundos. Cabe destacar, que no se hizo mayor hincapié en los tiempos en que incurría el proceso en su conjunto, debido a que éste no era un tema planteado como relevante, ya que la extracción y clasificación de información a partir de las páginas no es una tarea a realizarse en forma asidua.

Se obtienen para las entidades, los datos presentados en la Figura 23.

Entidades	FN	VP	FP	R	P	F-m [0.5]
organización	56	97	24	0,63	0,80	**0,71**
documento	11	42	13	0,79	0,76	**0,78**
evento	5	32	8	0,86	0,80	**0,83**
país	0	81	1	1,00	0,99	**0,99**
ciudad	4	24	1	0,86	0,96	**0,91**
persona	15	8	4	0,35	0,67	**0,46**
fecha	14	22	0	0,61	1,00	**0,76**
región	1	12	3	0,92	0,80	**0,86**
cargo	6	16	0	0,73	1,00	**0,84**
sigla	0	99	3	1,00	0,97	**0,99**
Total	**112**	**433**	**57**	**0,79**	**0,88**	**0,84**

Figura 23 - Resultados del reconocimiento de entidades

Al estimar los errores de reconocimiento de relaciones, se observó que los errores cometidos eran sensiblemente mayores al de reconocer entidades, por lo que se hizo un análisis más detallado, encontrándose que, en gran parte, ese aumento en el error era debido a los errores cometidos al reconocer las entidades. Por consiguiente, además de haberse estimado los errores reales, se estimaron los errores esperados al reconocer relaciones bajo la hipótesis de que no se cometen errores al reconocer las entidades. Los resultados obtenidos, se pueden observar en la Figura 24. Donde las columnas con asterisco (*) son los resultados esperados bajo la hipótesis antes mencionada.

Relaciones	FN	VP	FP	R	P	F-m[0.5]	FN*	VP*	FP*	R*	P*	F-m[0.5]
abreviación	10	20	4	0,67	0,83	**0,74**	5	25	4	0,83	0,86	**0,85**
pertenece a	24	23	8	0,49	0,74	**0,59**	5	42	8	0,89	0,84	**0,87**
vinculado a	5	4	9	0,44	0,31	**0,36**	2	7	4	0,78	0,64	**070**
ciudad de	5	24	1	0,83	0,96	**0,89**	1	28	0	0,97	1,00	**0,98**
persona cargo	2	3	0	0,60	1,00	**0,75**	2	3	0	0,60	1,00	**0,75**
Total	**46**	**74**	**22**	**0,62**	**0,77**	0,69	**15**	**105**	**16**	**0,88**	**0,87**	0,87

Figura 24 - Resultados del reconocimiento de relaciones

Considerando la aplicación en su conjunto, los errores cometidos al reconocer entidades y relaciones son:

recuperación: 0,76
precisión: 0.87
F-measure (0.5): 0,81

En la Figura 25, se presentan los resultados obtenidos al procesar la totalidad de los informes (1189), en un tiempo total de 2 horas 40 minutos.

Tipo de Instancia	Nº de ocurrencias reconocidas
organización	4236
documento	1140
evento	2936
país	8032
ciudad	2805
persona	1749
fecha	2364
región	1850
cargo	1608
sigla	9939
abreviación	1346
pertenece a	355
vinculado a	600
ciudad de	2176
persona cargo	644
Total	**41780**

Figura 25 - Resultados obtenidos por txt2xml

4.2.2 xml2owl

Los resultados obtenidos con esta aplicación para un *XML* de entrada (obtenido en la etapa anterior) de 36.659 entidades y 5.121 relaciones, así como información de la ontología base pueden verse en la Figura 26. El tiempo total insumido en esta etapa fue de 13 minutos 43 segundos.

Tipo de Instancia	Nº de ocurrencias en ontología base	Nº de ocurrencias en ontología generada
organización	349	1918
documento	0	443
evento	29	1016
país	205	205
ciudad	1372	1372
persona	0	1144
fecha	0	919
región	0	10
cargo	0	39
sigla	146	1351
abreviación	22	641
pertenece a	0	302
vinculado a	0	531
ciudad de	1317	1317
persona cargo	0	483
Total	**3440**	**11691**

Figura 26 - Datos de Ontología Base y Generada

La ontología generada consta de una cantidad de datos considerable, lo que no es suficiente si la calidad de los mismos no es por lo menos aceptable. En este caso, y en base a los resultados obtenidos en la etapa anterior (*text2xml)*, puede apreciarse que la calidad de los datos obtenidos es buena, lo que se transmite en forma directa a la calidad de los datos de la ontología generada.

Por consiguiente, la solución propuesta da como resultado una ontología que colma las expectativas previas, tanto cuantitativa, como cualitativamente.

Capítulo 5 OntoSearch

5.1 Introducción

El cometido de la aplicación *OntoSearch*[3] es mostrar algunas de las potenciales funcionalidades que posee la ontología representativa del sitio *Choike*, generada por el sistema *OntoChoike*.

Por ejemplo, encontrar la información que se obtuvo en la extracción de los términos relevantes y utilizarla para mejorar las funcionalidades que brinda el sitio Choike a sus lectores.

Por consiguiente, es el primer objetivo de la aplicación, brindar la posibilidad de llevar a la práctica el funcionamiento de un buscador sobre la ontología, para lo cual se diseñaron cuatro búsquedas.

Una búsqueda sencilla consiste en determinar las entidades que el sistema *OntoChoike* ha reconocido. Por ejemplo, indicar todas las entidades detectadas (personas, organizaciones, países, etc.) y en qué informes fueron encontradas.

Otra posibilidad es realizar una búsqueda por las relaciones que el sistema reconoce. Por ejemplo, buscar todas las entidades relacionadas con otra entidad o en cuáles informes se encuentra una relación dada.

Una tercera opción es la búsqueda por las categorías que presenta el sitio *Choike* por defecto. Como ejemplo de las mismas se puede citar gente, sociedad, ambiente, etc. que a su vez presentan subcategorías como ser género, pueblos indígenas, medio, discapacidades, etc.

Por último, obtener las instancias reconocidas y los informes que estén más relacionados con el informe original, siendo éste último un tema de discusión, ya que se deberá definir que se entiende por “informe más relacionado”.

Estas búsquedas son interesantes, pero lo que las hace más importantes aún es que no se pueden realizar en otros generadores de ontologías, como por ejemplo, *Protégé* no permite estas funcionalidades.

En cuanto a la implementación, se necesita un software que tenga la tarea de realizar la lógica de todas las búsquedas sobre la ontología generada previamente y provea de formas para visualizar los resultados que se obtengan de estas búsquedas.

A continuación se describe como se realizaron las búsquedas y que decisiones se tomaron para su resolución.

[3] Se recomienda ver Manual de Usuario del prototipo.

5.2 Solución propuesta

5.2.1 Búsqueda básica

La búsqueda más sencilla que brinda la aplicación (*Búsqueda básica – Ver* Figura 27) es encontrar las ocurrencias de las entidades en la ontología, junto con los informes en los cuales fueron efectivamente detectadas. Estos informes pueden a su vez ser visualizados por medio de un navegador Web.

Para realizar la búsqueda, se puede ingresar parte del texto de un término relevante, por ejemplo: el nombre o apellido de una persona, el nombre de una organización, evento o documento, una fecha, una ciudad, un país o una región (ver Figura 9).

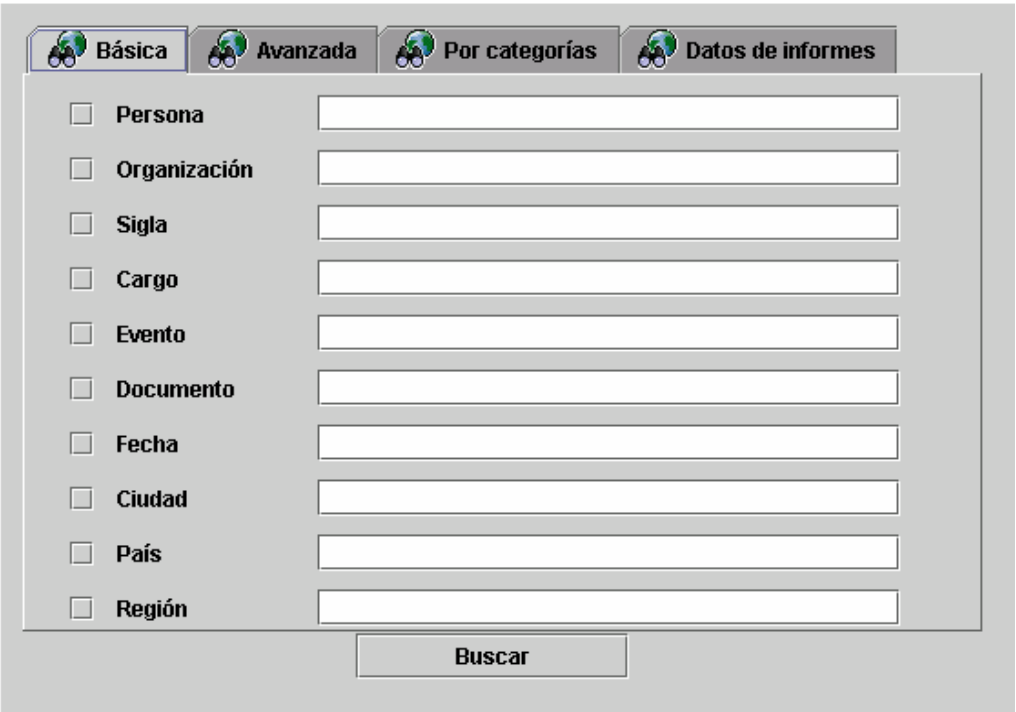

Figura 27 - OntoSearch

Para citar un ejemplo, si se ingresa el texto "*Vicente*" y se busca como persona, se obtendrá como resultado la tabla de la Figura 28.

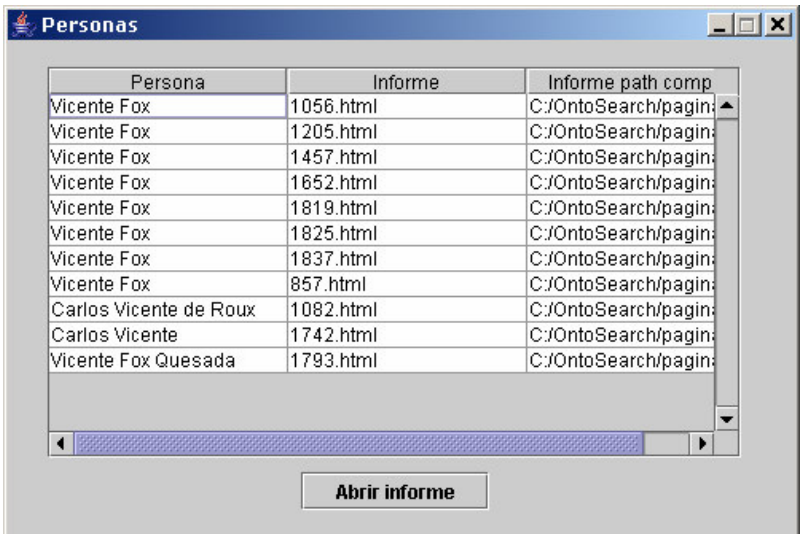

Figura 28 - Persona Vicente

Si además, se selecciona en la tabla la persona "*Vicente Fox*" o el informe correspondiente, y se desea abrirlo, la aplicación muestra una ventana con el documento en la cual se puede ubicar la persona nombrada (Ver Figura 29).

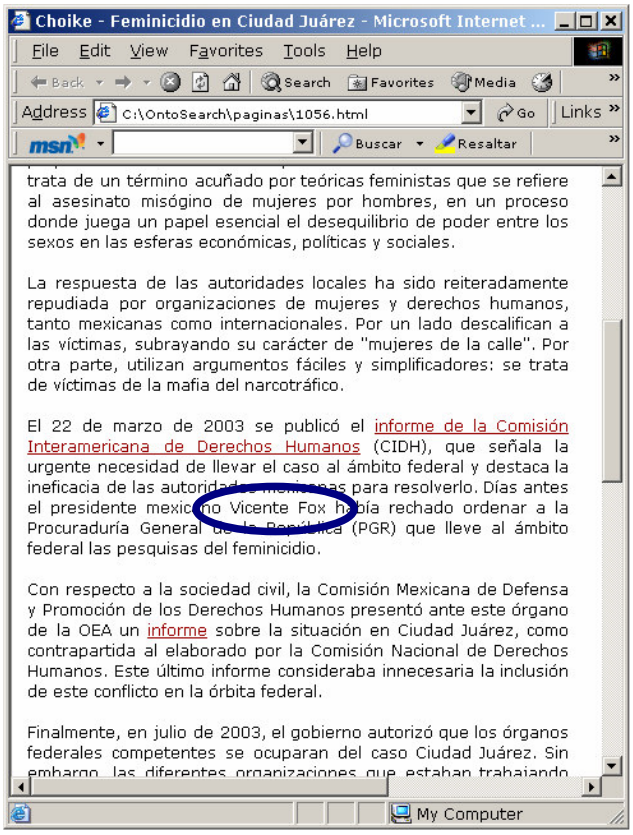

Figura 29 - Abrir informe que contiene "Vicente Fox"

5.2.2 Búsqueda avanzada

El sistema brinda además la posibilidad de una *búsqueda avanzada (Ver* Figura 30*)*. La misma permite encontrar las relaciones definidas como relevantes en la ontología.

Por ejemplo, si se desea buscar las organizaciones eventos o documentos asociados a la sigla "*BM*" se obtendrá "*Banco Mundial*" como se puede apreciar en la Figura 31.

El procedimiento es análogo al de la búsqueda básica, con la única diferencia que, en vez de buscar sobre las entidades, lo hace sobre las relaciones definidas en la ontología.

Estas búsquedas pueden llegar a realizarse utilizando *Protégé*, pero el inconveniente es que se deben configurar varios parámetros para lograr una búsqueda similar.

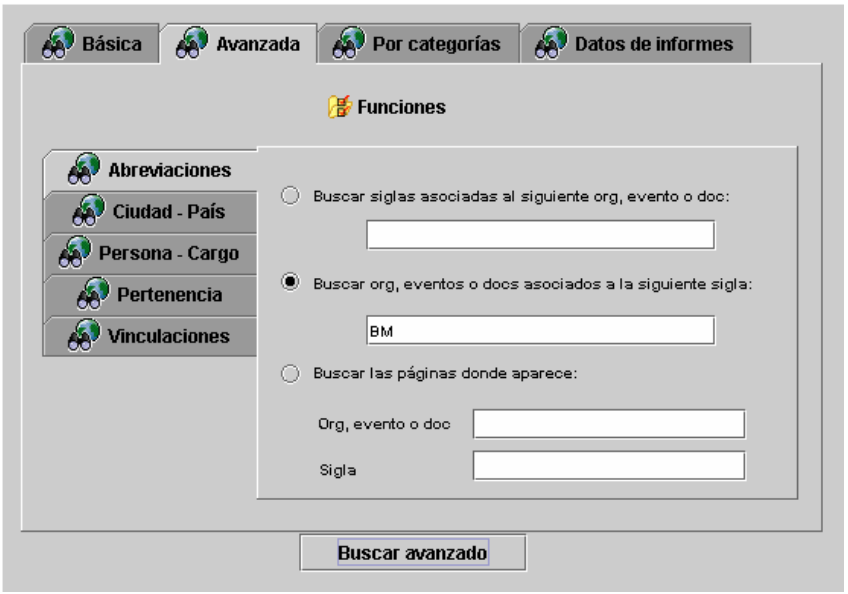

Figura 30 - Abreviaciones

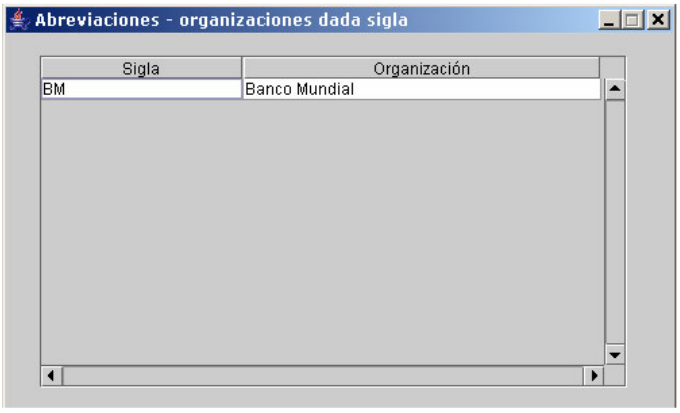

Figura 31 – Sigla "BM"

5.2.3 Búsqueda por categorías

La tercera búsqueda (por categorías – ver Figura 32) considera las categorías y subcategorías del sitio *Choike*, las cuales se pueden apreciar en la Figura 3.

En la aplicación, al seleccionar cierta categoría, se pueden ver las subcategorías que la componen y así poder realizar una búsqueda más afinada.

Además se presenta una opción que tiene el cometido de mostrar todos los informes pertenecientes a la propia categoría, sin diferenciarlos en subcategorías.

En la ontología se tiene la información de todos los informes que pertenecen a cada categoría y subcategoría. Entonces, la búsqueda en cierta categoría implica simplemente recuperar esta clase en la ontología y presentar los informes que la contienen.

Al igual que en las búsquedas anteriores, se brinda la posibilidad de abrir por medio de un navegador Web todos los informes que estén relacionados.

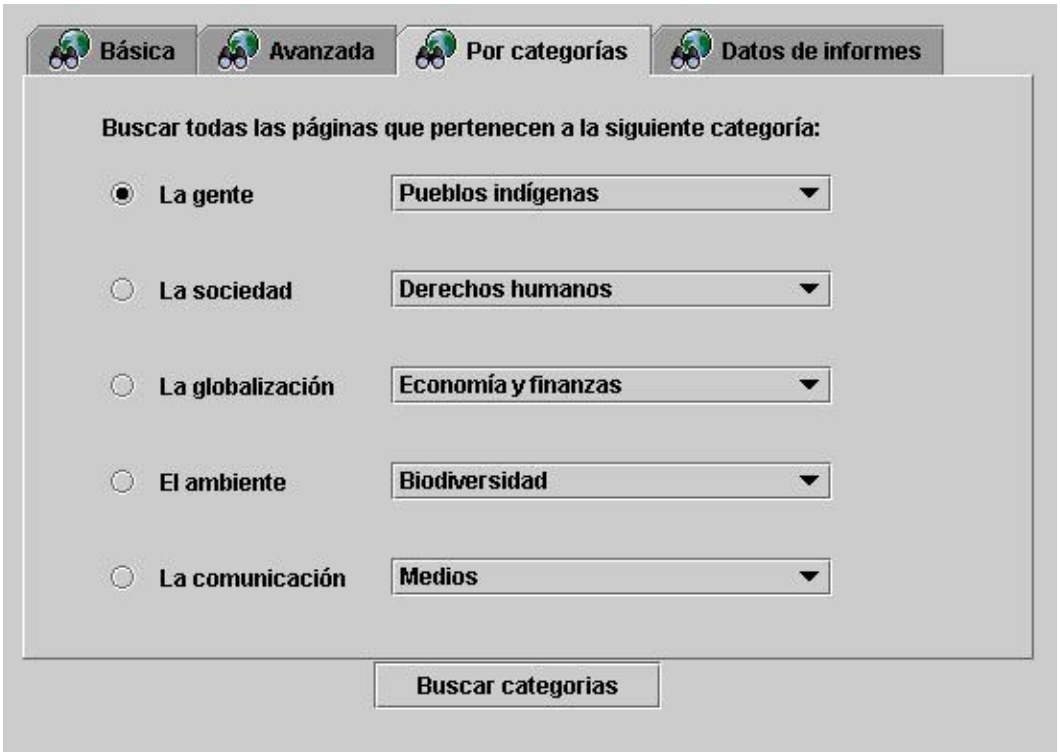

Figura 32 - Búsqueda por categorías

5.2.4 Datos de informes

Por último, se consideró interesante brindar la posibilidad de mostrar datos relevantes de un informe. (*Datos de informes – Ver* Figura 33).

La primera opción presenta todas las instancias reconocidas para un informe.

La segunda opción consiste en buscar las cinco entidades de la ontología que ocurren de modo más frecuente en un informe.

Para los dos casos anteriores, se brinda la posibilidad de abrirlos por medio de un navegador Web.

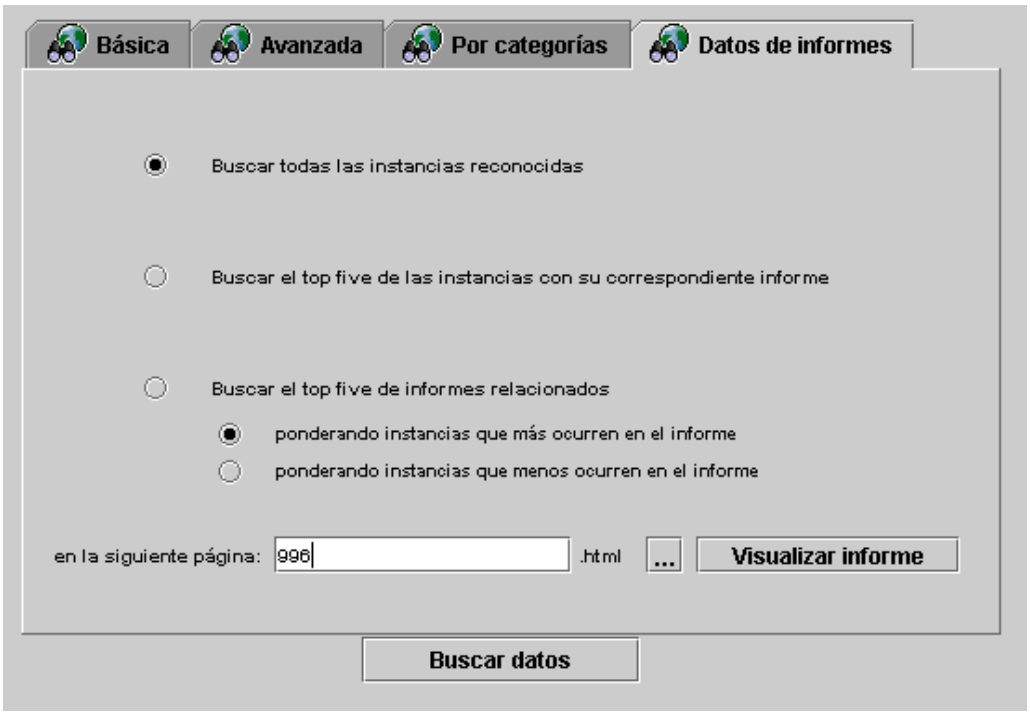

Figura 33 - Datos de un informe

La tercera opción presenta los cinco informes más relacionados con el informe original. Cabe aclarar que se entenderá como los más relacionados a aquellos documentos que muestren una mayor similitud en los temas sobre los que trata.

Pero ¿cómo saber cuál informe esta más relacionado con otro?. Antes que nada, se debe definir una función que "mida" el grado de similitud entre dos informes. Para responder a esta interrogante, se realizaron varias suposiciones.

En primer lugar, se tuvo en cuenta las entidades que se detectan en el informe original, y se busca aquellos informes que también las presentan. Cuantas más entidades del primer informe contengan, más deberían tender a hablar de los mismos temas.

Sin embargo, en un informe posiblemente haya más instancias de una ocurrencia que de otras: un informe que tenga las instancias más frecuentes será más similar que otro que sólo posea aquellas de más baja aparición en el original.

Por otra parte, también es probable que un documento que tenga instancias poco frecuentes sea más similar que otro. Las instancias que no aparezcan en todos los informes y son muy específicas pueden ser más "selectivas": si se encuentra otro que las contenga, es probable que traten de los mismos temas.

Por ejemplo, si en el informe original se encuentra la instancia "*Dikken Kloewicksz*", que no es un nombre muy común, es razonable pensar que cualquier otro que lo mencione tiene un fuerte vínculo con el primero.

En resumen, las dos opciones planteadas son las siguientes:

- Los cinco informes que presenten más cantidad de instancias del informe original en común, ponderando las que tengan más ocurrencias. Para este caso se calculó la función de la siguiente forma:

$$Similitud\ (X_1,X_2) = \sum_{i \in I_1} p_i$$

$$\text{donde } p_i = \begin{cases} \dfrac{\#\ de\ ocurrencias\ de\ la\ instancia\ i\ en\ X_1}{\#\ de\ ocurrencias\ totales\ en\ X_1} & , \text{si } i \in I_2 \\ 0 & , \text{en otro caso} \end{cases}$$

  Siendo:

  - X_1 el informe dado
  - X_2 otro informe
  - I_1 todas las instancias que ocurren en X_1
  - I_2 todas las instancias que ocurren en X_2

- Los cinco informes que presenten mayor cantidad instancias del informe original en común, ponderando las que tengan menos ocurrencias:

$$Similitud\ (X_1,X_2) = \sum_{i \in I_1} p_i$$

$$\text{donde } p_i = \begin{cases} \dfrac{1 + \#\ ocurren\ totales\ en\ X_1 - \#\ ocurren\ i\ en\ X_1}{1 + ocurrencias\ totales\ en\ X_1} & , \text{si } i \in I_2 \\ 0 & , \text{en otro caso} \end{cases}$$

Con estas opciones no se agotan las posibilidades. Se puede, por ejemplo, tomar como criterio el ponderar las ocurrencias de cierta clase o relación particular (sólo las organizaciones, sólo las personas, etc.), bajo el supuesto que determinan mejor el tema del informe. Existen miles de combinaciones a ensayar para este prototipo, pero por una cuestión de tiempo, sólo se implementan las dos anteriores.

Cabe notar que si la cantidad de ocurrencias totales en el informe original es cero, su similitud respecto cualquier otro informe también es cero.

Para su mejor comprensión se presentará un ejemplo. En la Figura 34 se muestran las entidades reconocidas para cada informe (entre paréntesis, el número de instancias presentes en cada documento).

Informe 1	Informe 2	Informe 3
Juan (1)	Pedro (1)	José (1)
María (4)	María (1)	Pedro (1)
Pedro (1)	José (1)	Sofía (1)

Figura 34 – Ejemplo

Para la primera opción, si el informe original es "*Informe 1*":

Similitud ^{op1} (*Informe 1*, *Informe 2*) = $p_{Juan} + p_{Maria} + p_{Pedro}$

donde $p_{Juan} = 0$ porque *Juan* no pertenece al *Informe 2.*

$p_{Maria} = \frac{\#\ ocurr\ \ instancia\ Maria\ en\ Informe_1}{\#\ ocurr\ totales\ en\ \ Informe_1} = \frac{4}{6}$, *Maria* fue reconocida en 2.

$p_{Pedro} = \frac{\#\ ocurr\ \ instancia\ Pedro\ en\ Informe_1}{\#\ ocurr\ totales\ en\ \ Informe_1} = \frac{1}{6}$, *Pedro* fue reconocido en 2.

Como resultado se obtiene: Similitud (*Informe 1*, *Informe 2*) = 5/6.

Similitud ^{op1} (*Informe 1*, *Informe 3*) = $p_{Juan} + p_{Maria} + p_{Pedro}$

donde $p_{Juan} = 0$ porque *Juan* no pertenece al *Informe 3.*

$p_{Maria} = 0$ porque *Maria* no pertenece al *Informe 3.*

$p_{Pedro} = \frac{\#\ ocurr\ \ instancia\ Pedro\ en\ Informe_1}{\#\ ocurr\ totales\ en\ \ Informe_1} = \frac{1}{6}$, *Pedro* fue reconocido en 3.

Entonces se obtiene: Similitud ^{op1} (*Informe 1*, *Informe 3*) = 1/6.

En conclusión se puede apreciar que la similitud entre el *Informe 1* y el *Informe 2* es mayor, lo cual es razonable ya que se esta ponderando por la cantidad de ocurrencias del informe original (*Informe 1*) y *María* aparece cuatro veces.

Para la segunda opción, si el informe original es "*Informe 1*":

Similitud ^{op2} (*Informe 1*, *Informe 2*) = $p_{Juan} + p_{Maria} + p_{Pedro}$

donde $p_{Juan} = 0$ porque *Juan* no pertenece al *Informe 2.*

$$p_{Maria} = \frac{1 + \#\text{ oc totales en } Info_1 - \#\text{ oc Maria en } Info_1}{1 + \text{oc totales en } Info_1} = \frac{1 + 6 - 4}{1 + 6} = \frac{3}{7}$$

$$p_{Pedro} = \frac{1 + \#\text{ oc totales en } Info_1 - \#\text{ oc Pedro en } Info_1}{1 + \text{oc totales en } Info_1} = \frac{1 + 6 - 1}{1 + 6} = \frac{6}{7}$$

Como resultado se obtiene: Similitud ^{op2} (*Informe 1*, *Informe 2*) = 9/7.

Similitud ^{op2} (*Informe 1*, *Informe 3*) = $p_{Juan} + p_{Maria} + p_{Pedro}$

donde $p_{Juan} = 0$ porque *Juan* no pertenece al *Informe 3.*

$p_{Maria} = 0$ porque *Maria* no pertenece al *Informe 3.*

$$p_{Pedro} = \frac{1 + \#\text{ oc totales en } Info_1 - \#\text{ oc Pedro en } Info_1}{1 + \text{oc totales en } Info_1} = \frac{1 + 6 - 1}{1 + 6} = \frac{6}{7}$$

Entonces se obtiene: Similitud ^{op2} (*Informe 1*, *Informe 3*) = 6/7.

En conclusión se puede apreciar que la similitud entre el *Informe 1* y el *Informe 2* es mayor, porque si bien el reconocer a la instancia *Maria* tiene poco peso por presentar tantas ocurrencias en el *Informe 1*, *Pedro* hace que la similitud sea más grande.

Ahora, si se supone que *Pedro* no pertenece al *Informe 2* la similitud entre el *Informe 1* y el 2 ya no es mayor porque *Pedro* aparece menos veces que *Maria* y cabe recordar que esta opción pondera los que tienen menos ocurrencias.

Dado que se implementaron únicamente las dos primeras opciones, no se puede afirmar que sean las más acertadas. Los casos de combinaciones (tercera opción) podrían dar mejores resultados.

En la Figura 35, se puede apreciar cómo se obtienen los más relacionados y de que forma se visualiza un informe y los resultados de las posibles búsquedas.

Figura 35 - Visualización de un informe

En el sector izquierdo se muestran los colores que representan a las entidades, las relaciones reconocidas en el informe en cuestión, las primeras cinco instancias que más aparecen y los primeros cinco informes que estén más relacionados con el informe dado.

Por otro lado, el sector derecho contiene el informe, en donde las ocurrencias de las entidades reconocidas se marcan con colores, de acuerdo a su clase (persona, documento, evento, etc.).

En caso de que la búsqueda contenga informes, se brinda además la posibilidad de visualizarlos de la misma forma que al original.

5.3 Implementación

Para la resolución del buscador se crearon una página y cuatro clases fundamentales, teniendo cada una de ellas una tarea bien definida.

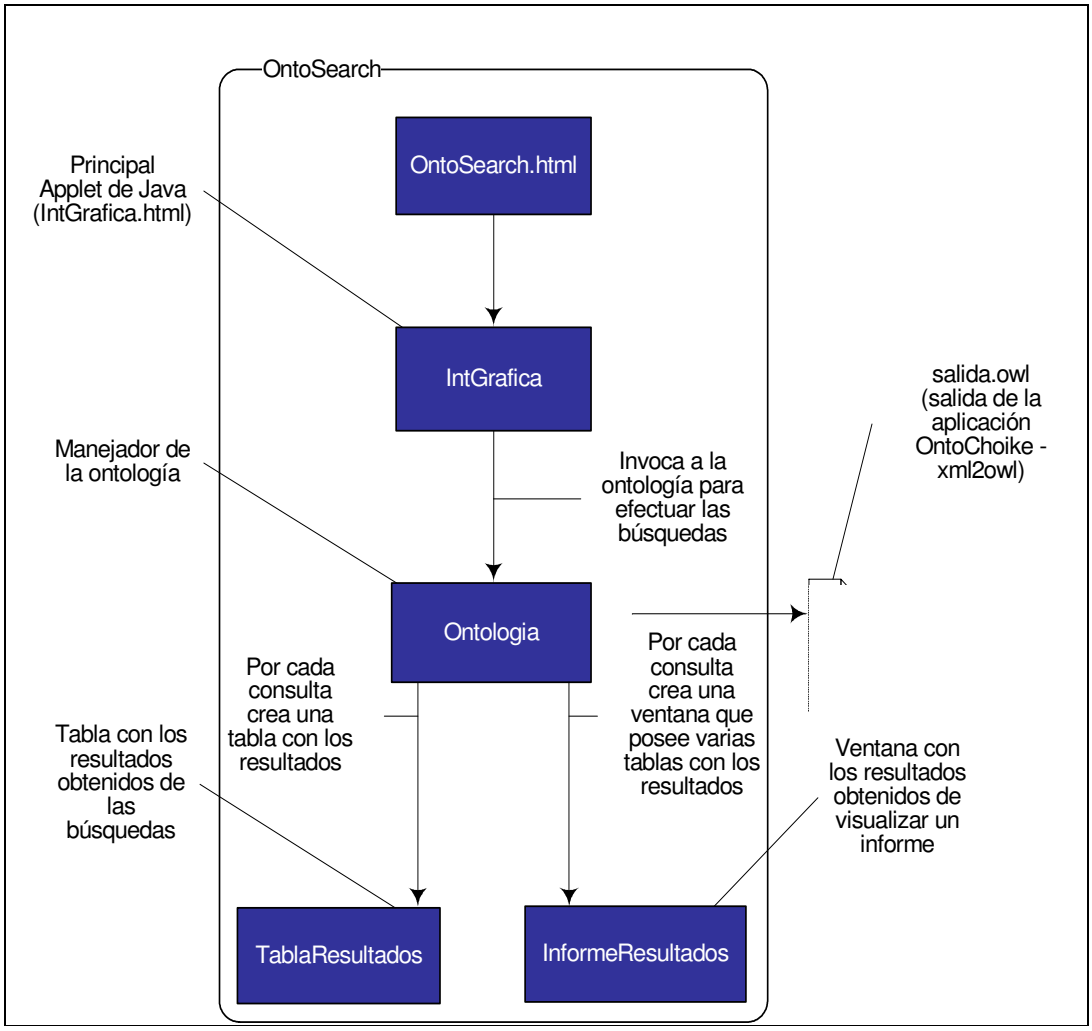

Figura 36 - Arquitectura de la aplicación

Como se puede apreciar en la Figura 36, la arquitectura del sistema consiste en:

- *OntoSearch.html*
  La página encargada del ingreso de los parámetros y de la carga del *Applet* con las búsquedas.

- *IntGrafica*
  Es el propio *Applet* que tiene el cometido de presentar las búsquedas y de invocar a las funciones de la ontología.

- *Ontologia*
  Se encarga de hacer efectivas las búsquedas utilizando la ontología que genera el sistema *OntoChoike*.

- *TablaResultados*
  Presenta la tabla con los resultados de las búsquedas y la posibilidad de abrir los informes.

- *InformeResultados*
  Presenta la visualización de las entidades y relaciones reconocidas y marcadas en el propio informe, junto a las instancias y documentos más relacionados.

La página *OntoSearch.html* es simplemente la encargada de enviar al sistema ciertos parámetros y abrir la página *IntGrafica.html* que tiene el cometido de cargar el *Applet* principal.

Para la implementación de la aplicación fue necesario el ingreso de dos parámetros consistentes en la dirección del archivo que contiene la ontología (*salida.owl*) y el directorio donde se encuentran las páginas del sitio. A continuación, se despliega en pantalla el sistema como se apreció anteriormente en la Figura 27.

La implementación del sistema *OntoSearch* se realizó con un *Applet* de *Java* para lograr que la aplicación sea multi plataforma.

Para las búsquedas sobre la ontología se utilizó un paquete de *Java* que implementa un lenguaje (basado en *XML*) permitiendo seleccionar subconjuntos de un documento *XML*. La idea es parecida a las expresiones regulares para seleccionar partes de un texto plano. Además, permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del *XML*. Este paquete es denominado como *XPath* [37].

Otra clase que se implementó fue la clase *Ontología*, la cual tiene como primera tarea levantar por medio del *JDom* [36], el documento con toda la ontología (que se encuentra en el archivo *salida.owl*), para así poder utilizarlo en cualquier búsqueda futura que se presente.

La clase *TablaResultados* tiene el cometido de presentar los resultados de todas las búsquedas por medio de una tabla y en caso que contenga informes se brinda la posibilidad de abrirlos por medio de un navegador Web.

Para la visualización de la solución se utiliza un *JTable* en un *JFrame* de *Java* el cual contiene los resultados de la búsqueda.

Por último, por medio de la clase *InformeResultados* se visualiza un informe y los resultados de las posibles búsquedas sobre él mismo.

Para la visualización de la solución se utilizan varios *JTable* y un *JEditorPane* en un *JFrame* de *Java* el cual contiene los resultados de la búsqueda.

5.4 Evaluación

5.4.1 Introducción

Lamentablemente, debido al tiempo y recursos con los que cuenta este proyecto, no es posible realizar un análisis profundo de los resultados obtenidos por *OntoSearch*. Sin embargo, es interesante realizar algunas consideraciones sobre los resultados obtenidos en la búsqueda de los informes relacionados.

5.4.2 Forma de evaluación

Para poder realizar la estimación de cómo se comporta el sistema en la búsqueda de informes relacionados, se redujo el problema considerando un conjunto de veinte documentos tomados de forma aleatoria. Para cada uno de ellos se obtienen los cinco más relacionados totalizando cien informes como resultado.

Luego, se analizan los resultados, verificando si son coherentes con los temas presentados en cada uno de ellos.

Cabe aclarar que esta discusión se centra en el criterio personal de los desarrolladores de ese prototipo, dado que no se cuenta con una medida objetiva para determinar el grado de similitud entre dos informes. Además, se desconoce cuáles de todos los informes de *Choike* son efectivamente (y de forma absoluta) los más relacionados a un documento dado.

5.4.3 Conclusiones

Si se selecciona la primera regla de cálculo de similitud, en la gran mayoría de los casos, todos los informes obtenidos trataban de temas similares al informe en cuestión, y tan sólo siete no tenían ninguna relación con el tema. Se observó que muchos de los informes incorrectos fueron seleccionados por contener una organización en común.

Por ejemplo, para uno de la categoría *Paz y seguridad mundial,* que habla sobre los medios de Medio Oriente, se presenta como resultado informes de la misma categoría o de la categoría *Derechos humanos* que hablan sobre la prensa Palestina y el Medio Oriente, el periodismo, violaciones a los derechos humanos en el terrorismo, guerra, ocupación en Irak, etc.

Eso para citar un ejemplo, pero se ha visto que los informes relacionados en su mayoría, o bien están en la misma categoría de *Choike* que el primero, o bien en otra categoría, pero su tema es muy similar.

La segunda regla presenta sus sorpresas, ya que sus resultados son muy cercanos a la primera opción: en la mayoría de casos da como resultado informes que se encuentran en la primera opción pero con distinto orden. Aunque varios no tienen relación con el primero, son muy pocos[4] comparados con la cantidad que se obtuvieron como resultados.

[4] De cien informes solo diecisiete no tienen relación con el informe original.

En resumen, y a criterio personal, se considera que la primera opción es la más acertada, aunque cabe destacar que la segunda también genera resultados razonables.

Finalmente, por más que estos resultados son satisfactorios, no hay que olvidar que se está frente a una pequeña evaluación: sólo se tomaron veinte informes de los aproximadamente mil doscientos existentes.

Capítulo 6 Conclusiones y trabajo futuro

En este capítulo, se analizan las conclusiones obtenidas como resultado del proyecto, distinguiéndose las conclusiones relativas al problema, a la tecnología empleada y los aportes obtenidos. Se presentan luego las posibles mejoras y extensiones a realizar.

6.1 Conclusiones sobre el problema

En el presente proyecto se presenta una solución incremental al problema planteado de la generación de anotación semántica para páginas Web y generación semiautomática de ontologías.

Se implementó para este fin un sistema (*OntoChoike*) que genera una ontología para el sitio *Choike* utilizando el estándar *OWL*, el cual permite la interacción con múltiples herramientas. Este sistema, por medio de técnicas de extracción de información, detecta entidades y relaciones en las páginas de *Choike*, volcándolas a la ontología.

Basado en la ontología generada, se implementó un prototipo (*OntoSearch*) que demuestra las ventajas que se logran al disponer de ontologías específicas para un sitio *Web*, a la hora de hacer búsquedas simples o complejas, encontrar documentos relacionados, etc.

En resumen, se encontró la presentación de una posible solución a un problema complejo mediante la utilización de técnicas de extracción para un escenario *Web*, la implementación de un prototipo que refleja una solución concreta al problema planteado, la utilización de herramientas difundidas dentro del área y brindar información que permite seguir trabajando para mejorar la solución planteada. Se cumplieron, entonces, todos los objetivos que se tenían planteados, tanto en el prototipo *OntoChoike* como en el *OntoSearch*.

6.2 Conclusiones sobre las tecnologías empleadas

Este proyecto tuvo una importante dedicación en la investigación y estudio de estándares (*XML*, *DAML-ONT*, *DAML-OIL*, *RDF Schema*, *UML*, etc.), así como de herramientas para la generación de ontologías.

El uso de estándares para la representación e intercambio de información como *XML* y *OWL (XML* para el almacenamiento de la información extraída y *OWL* para el almacenamiento de la ontología generada) permitió el uso de herramientas como *JDOM* o *Protégé*, simplificando los problemas a resolver.

Como editor de ontologías, el hecho de haber utilizado *Protégé*, facilitó la creación de la ontología base, así como también su visualización, siendo de gran utilidad a la hora de encontrar los errores que se cometían al generarla.

Finalmente, en cuanto al aspecto puramente tecnológico, el utilizar *Java* como lenguaje base resultó ser una buena elección a la hora de integrar diversas herramientas a fin de resolver los problemas que se presentaron en las distintas etapas que conforman la solución.

6.3 Aportes a los desarrolladores

Uno de los aportes más importante de este proyecto fue la experiencia adquirida en las tecnologías estudiadas, y el entendimiento de los conceptos relacionados al mundo de los *metadatos*, y ontologías en general, llegando a entender los diferentes niveles de abstracción que se encuentran en cada uno, y las diferentes posturas que existen para generar una ontología y trabajar con la misma.

No menos importante fueron los aportes en el área de extracción y clasificación de información, donde se vieron los problemas que se encuentran con asiduidad en esta tarea y las distintas alternativas de las que se dispone para resolverlos.

6.4 Trabajos futuros

A medida que se realizo el proyecto surgieron varios puntos a tener en cuenta para los trabajos futuros: mejoras al sistema OntoChoike, en el buscador y en la forma de relacionar informes.

6.4.1 Mejoras al sistema OntoChoike

Para el sistema *OntoChoike* se presentarán algunas ideas propuestas, habiendo adicionalmente una gran variedad de modificaciones. Por ejemplo, se puede citar las posibles mejoras a las reglas de inferencia para reconocer otras entidades y relaciones.

Por otro lado, en el sistema actual, el usuario tiene poca capacidad de decisión sobre el proceso de extracción. Algunas posibles opciones tendientes a solucionar este problema son las siguientes:

- Poder seleccionar de las entidades y relaciones ya existentes en los diccionarios cuáles se desea extraer dinámicamente en el momento de la generación.

- Brindar la posibilidad de ampliar los diccionarios que utiliza el sistema, ya sea de forma manual o semiautomática, para así poder reconocer nuevas entidades y relaciones.

6.4.2 Buscador

Otro trabajo interesante se centra en la posibilidad de realizar un buscador avanzado para el portal sobre la ontología que genera *OntoChoike*: actualmente sólo se dispone de una organización en categorías y de un buscador que permite rastrear información en el sitio.

Debido al hecho de que *Choike* es un portal destinado a mejorar la visibilidad de los contenidos producidos por las *ONG* y que actúa como una plataforma donde éstas pueden difundir su trabajo y a su vez alimentarse de diversas fuentes de información, es que resulta de vital importancia tener un buscador avanzado.

Sería recomendable mejorar el acceso a la información, explotando los *metadatos* y la ontología generados, para realizar un sistema de búsqueda avanzada y de consultas que apuntaran a una mayor satisfacción del usuario del sitio. Por citar un ejemplo, se podría presentar en las propias páginas de *Choike* los informes más relacionados como sugerencia de posibles lecturas.

6.4.3 Investigar forma de relacionar dos informes

Finalmente, se considera necesario investigar formas alternativas de determinar los informes más relacionados.

En la implementación de *OntoSearch* se tomaron en cuenta tan solo dos posibilidades de las diversas que se pueden utilizar. Sería interesante ver otras y comparar sus resultados con los obtenidos en el prototipo.

Entre estas opciones está la de considerar las entidades, relaciones y el número de ocurrencias de las mismas en los informes. Se puede, por ejemplo, implementar la función similitud dándole más peso a las organizaciones o personas y menos peso a las fechas, ya que dos fechas similares no siempre reflejan temas similares.

Otra posibilidad es considerar la categoría a la que pertenecen las entidades y relaciones. Es razonable pensar que dos informes en una misma categoría presentan una mayor probabilidad que dos en categorías diferentes.

Bibliografía

[1] Web Semántica
http://www.w3.org/DesignIssues/Semantic.html
Fecha de acceso: 11/05/2004

[2] Web Semántica
http://www.w3.org/2001/sw/
Fecha de acceso: 11/05/2004

[3] Gruber, T. R. A translating approach to portable ontology specifications. *Knowledge Acquisition*, 1993, 5, pp. 199-220.

[4] Taylor, A. *The Organization of Information*. Englewood, Colorado: Libraries Unlimited, 1999.

[5] N. F. Noy, D. MacGuiness. *Ontology development 101: A guide to creating your first ontology* – Standford Knowledge Systems Laboratory. 2001

[6] Luke, S.; Spector, L.; Rager, D. (1996). Ontology-Based Knowledge Discovery on the World-Wide Web. http://www.cs.umd.edu/projects/plus/SHOE/pubs/aaai-paper.html

[7] SHOE.
http://www.cs.umd.edu/projects/plus/SHOE/
Fecha de acceso: 24/05/2004

[8] Fensel, D. et al. (2000). OIL in a Nutshell.
www.cs.vu.nl/~ontoknow/oil/downl/oilnutshell.pdf
Fecha de acceso: 24/05/2004

[9] W3C – Estándar RDF-Schema.
http://www.w3.org/TR/rdf-schema/
Fecha de acceso: 24/05/2004

[10] OWL Web Ontology Language Overview.
http://www.w3.org/2001/sw/WebOnt/TR/STAGE-owl-features/
Fecha de acceso: 24/05/2004

[11] Kent, Robert. Conceptual Knowledge Markup Language: The Central Core. TOC (The Ontology Consortium). http://sern.ucalgary.ca/ksi/kaw/kaw99/papers/Kent1 /CKML.pdf

[12] XOL.
http://www.ai.sri.com/pkarp/xol/
Fecha de acceso: 30/05/2004

[13] CycL.
http://www.cyc.com/cycl.html
Fecha de acceso: 24/05/2004

[14] McGuinness, Deborah; Fikes, Richard; Stein, Lynn; Hendler, James. DAML-ONT: An Ontology Language for the Semantic Web.

[15] Horrocks, Ian. DAML+OIL Technical Detail. Charla dada en W3C Web Ontology Working Group meeting, Bell Laboratories, Murray Hill NJ, January, 2002.

[16] KIF.
http://logic.stanford.edu/kif/kif.html
Fecha de acceso: 24/05/2004

[17] EBXML.
http://www.ebxml.org
Fecha de acceso: 24/05/2004

[18] F-Logic.
http://www.informatik.unifreiburg.de/~dbis/floridhttp://www.daml.org/2000/10/daml-ont.html
Fecha de acceso: 24/05/2004

[19] GRAIL.
http://www.opengalen.org/open/CRM/
Fecha de acceso: 24/05/2004

[20] OCML.
http://kmi.open.ac.uk/projects/ocml/
Fecha de acceso: 24/05/2004

[21] UML.
http://www.omg.org/technology/documents/formal/uml.htm
Fecha de acceso: 24/05/2004

[22] Gruber, T. R. Ontolingua: A mechanism to support portable ontologies. Knowledge Systems Laboratory, Noviembre de 1992.

[23] McGuinness. The Chimaera Ontology Environment. Knowledge Systems Laboratory, Stanford University, 2000

[24] OILED.
http://www.ontoknowledge.org/oil
Fecha de acceso: 24/05/2004

[25] Angus Roberts. An Introduction to OilEd.
http://www.cs.man.ac.uk/~horrocks/Teaching/cs646/OilEdTutorial-Ver1.1/
Fecha de acceso: 24/05/2004

[26] Protégé
http://protege.semanticweb.org/
Fecha de acceso: 11/10/2004

[27] VerticalNet
http://www.verticalnet.com/
Fecha de acceso: 11/10/2004

[28] RDF Schema
http://paranormal.se/perl/proj/rdf/schema_editor/___welcome.html
Fecha de acceso: 11/10/2004

[29] OIL
http://img.cs.man.ac.uk/oil /
Fecha de acceso: 11/10/2004

[30] OntoKnowledge
http://www.ontoknowledge.org/
Fecha de acceso: 11/10/2004

[31] OntoAgents
http://www-db.stanford.edu/OntoAgents/
Fecha de acceso: 11/10/2004

[32] Racer
http://www.racer-systems.info/
Fecha de acceso: 11/10/2004

[33] Protégé
http://protege.stanford.edu/plugins/owl/
Fecha de acceso: 11/10/2004

[34] WonderWeb
http://wonderweb.man.ac.uk/
Fecha de acceso: 11/10/2004

[35] HTMLParser
http://www.htmlparser.org/
Fecha de acceso: 13/03/2005

[36] JDOM
http://www.jdom.org/
Fecha de acceso: 13/03/2005

[37] XPATH
http://www.w3.org/TR/xpath
Fecha de acceso: 1/04/2005

Glosario

Agente
Persona que profesionalmente gestiona por cuenta ajena, mediante comisión, operaciones de venta u otras transacciones.

Aplicación
Programa preparado para una utilización específica.

Categorías
Jerarquía de clases establecidas para cierto tema. Cada una de las clases establecidas en una profesión, carrera o actividad.

Clustering
Técnica de análisis de datos en la que se agrupan las observaciones según su similitud. Es la agrupación que realizan por ejemplo los buscadores para no mostrar más de un cierto número de páginas de una Web para una determinada búsqueda.

Descriptor
Término o símbolo válido y formalizado que se emplea para representar inequívocamente los conceptos de un documento o de una búsqueda.

Expresiones regulares
Una expresión regular es un patrón que describe un conjunto de cadenas de caracteres. Es una forma de representar a los lenguajes regulares (finitos o infinitos) y se construye utilizando caracteres del alfabeto sobre el cual se define el lenguaje (usando operadores - unión, concatenación y clausura de *Kleene* - para combinar expresiones más pequeñas).

Freeware
Software de libre distribución, al que no siempre se tiene acceso a su código fuente y que por lo general no puede ser modificado.

Inferir
Sacar una consecuencia o deducir algo de otra cosa.

Interoperabilidad
Capacidad de comunicación entre diferentes programas y máquinas de distintos fabricantes.

Metadatos
Información referente a los datos, como ser, su ubicación, estructura, etc.

Modelo conceptual
Muestra gráficamente los conceptos (clases de objetos), los atributos y las asociaciones más importantes del dominio de un problema. Es una representación ideal que busca explicar un fenómeno.

Plugins
Un *plugin* (o *plug-in*) es un programa que interactúa con otro programa para aportarle una función o utilidad específica, generalmente muy específica.

Prototipo
Modelo a escala o facsímil de lo real, pero no tan funcional para que equivalga a un producto final, ya que no lleva a cabo la totalidad de las funciones necesarias del

sistema final. Proporcionando una retroalimentación temprana por parte de los usuarios acerca del Sistema.

Sistema
Una colección de unidades conectadas entre sí, que están organizadas para llevar acabo un propósito específico. Un sistema puede describirse mediante uno o más modelos, posiblemente desde puntos de vista distintos.

Taxonomías
Ciencia que trata de los principios, métodos y fines de la clasificación. Se aplica en particular, dentro de la biología, para la ordenación jerarquizada y sistemática, con sus nombres, de los grupos de animales y de vegetales.

UML
Unified Modeling Language es un lenguaje estándar definido por la OMG para análisis y diseño orientado a objetos.

XML
Extensible Markup Language es una especificación que define una forma estándar de agregar marcas a un documento, es en definitiva un lenguaje para documentos que contienen información estructurada.

XSL
Extensible Stylesheet Language es un lenguaje para expresar documentos de estilos. Un XSL es un archivo que describe cómo desplegar un documento XML de un tipo dado.

XSLT
Extensible Stylesheet Language Translator es un lenguaje que permite, dado un documento XML, generar otro, que por ejemplo, puede tener la misma información del inicial, pero con distinta distribución.