Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Sistema de búsqueda y recuperación de ontologías

Alan Christopher Alamillo Medina

Matrícula: 208333483

Asesores

Dra. Maricela Claudia Bravo Contreras

Profesor Asociado

Departamento de Sistemas

Dr. José Guadalupe Rodríguez García

Departamento de Computación del CINVESTAV

Unidad Zacatenco

Noviembre 2013

Tabla de Contenidos

Resumen 3

Introducción 3

Objetivo general 4

Objetivos específicos 4

Diseño 4

- Módulo: Araña Web 6
  - Sub-Módulo: Raspador (*scraper*) 6
  - Sub-Módulo: Análisis sintáctico (parser) 7
- Módulo: buscador Web 8
  - Sub-Módulo: Cálculo del factor de relevancia 9
  - Sub-Módulo: Analizador de relevancia 9

Implementación 10

- Software, APIs y otras herramientas 10

Pruebas del sistema 11

- Pruebas de la araña Web 11
- Pruebas del módulo de búsqueda (consola) 11

Bibliografía 14

Resumen

El conocimiento humano se ha ido manifestando de diferentes formas a lo largo de la historia. En los tiempos antiguos se utilizaban códices y libros para transmitir información entre personas. De esa manera, muchas de las cosas que conocemos hoy, tienen vigencia y relevancia, ya que el conocimiento que se adquirió en el pasado, ha sido de gran ayuda en el presente.

Con la llegada de Internet como un medio masivo de comunicación, se rompieron muchos paradigmas, y se logró por primera vez, contar con un canal de comunicación accesible y relativamente libre de censura, mediante el cual, las personas podrían intercambiar información (e inevitablemente, conocimiento), en tiempo real y con costos mínimos en términos de esfuerzo y dinero.

A medida que Internet maduró, se comenzaron a concebir diferentes herramientas que permitiesen una gestión eficiente y ordenada del conocimiento. Entre muchos conceptos que salieron a flote, se encuentran las ontologías.

La realización de este proyecto terminal permite la obtención de archivos en formato OWL para su posterior consulta mediante una interfaz Web. En un escenario típico de uso, un usuario podría ingresar a una interfaz Web mediante una URL específica. Una vez en dicha URL, se podrá hacer uso de un formulario que contiene un campo de texto; en dicho campo, el usuario ingresará palabras clave que permitirán la identificación de ontologías que coincidan con el criterio de búsqueda.

Introducción

La Internet se ha convertido en un canal de comunicación masivo, por medio del cual millones de personas interactúan todos los días. Como resultado de esta actividad, se generan cantidades enormes de información de índole y contexto diferente.

Con un volumen tan grande de información disponible, es difícil organizar y clasificar los datos de una manera que sea relevante para cualquier individuo que se encuentre recopilando información. Partiendo de dicha situación, se han desarrollado diferentes métodos de encontrar información que sea semánticamente relevante con lo que el usuario desea.

Una de esas herramientas es llamada ontología, la cual realiza una función de directorio en donde se almacenan colecciones de términos y sus relaciones. El uso de ontologías permite obtener información semántica, la cual es procesada por máquinas, generalmente usando técnicas de PLN (Procesamiento de Lenguaje Natural).

En esta propuesta se detallarán las especificaciones de una araña web (también conocido como web c*rawler*, o simplemente c*rawler*), así como un buscador de ontologías, que permita generar una lista de resultados relevantes para el usuario, usando una serie de palabras clave para determinar el contexto semántico de la información devuelta por el buscador.

Las ontologías que serán recuperadas por el buscador se referirán al contexto definido por las palabras clave proporcionadas por el usuario.

Objetivo general

Diseñar e implementar una aplicación web capaz de buscar y recuperar ontologías a partir de un conjunto de palabras clave ingresadas por una persona.

Objetivos específicos

- Diseñar e implementar una araña o *crawler* capaz de visitar sitios web y extraer referencias a otros documentos dentro del sitio visitado.
- Crear una colección ordenada de documentos de texto que se almacenarán en forma local para su consulta periódica.
- Programar un módulo para el análisis sintáctico de ontologías, también llamado parser de ontologías, con la finalidad de obtener información clave .
- Diseñar e implementar un sistema de cálculo de relevancia de ontologías que permita evaluar la calidad de los resultados devueltos por el buscador.
- Diseñar e implementar una aplicación web que permita desplegar los resultados obtenidos por el buscador.

Diseño

El sistema se conforma de tres módulos principales que aseguran la funcionalidad mínima necesaria. Ver *Figura 1*.

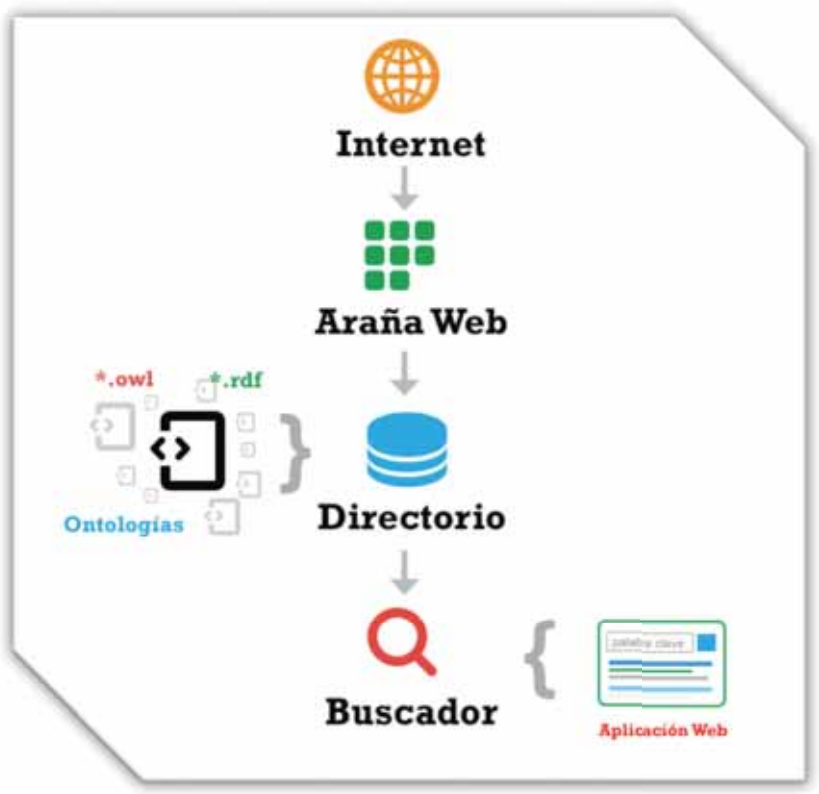


Figura 1. Esquema general del istema

I. *Crawler* o araña web. Este módulo se encargará de visitar páginas web específicas para extraer ontologías, que posteriormente se almacenarán de forma local. El almacenamiento se hará en forma de archivos con formato OWL o RDF

   De manera adicional, el módulo de la araña web contará con un componente clave: un analizador sintáctico (conocido también como *parser*). Dicho analizador sintáctico se encargará de extraer metainformación de cada ontología, como por ejemplo, la URL de procedencia y las clases relacionadas a determinada ontología

   Es importante especificar que mediante el análisis sintáctico se extraerá la metainformación antes mencionada para que esta sea almacenada en una base de datos local, en donde solo existirá metainformación de ontologías.

II. Directorio de ontologías. Se trata de un módulo que permitirá el almacenamiento de ontologías que han sido extraídas por la araña web. Este directorio servirá como fuente de información para el buscador de ontologías. Dicho directorio estará compuesto por archivos en formato OWL o RDF, y no contendrá información adicional.

III. Buscador. Este módulo tiene la tarea de realizar consultas al directorio de ontologías en base a palabras clave específicas, proveídas por el usuario.

Módulo: Araña Web

La araña Web es un componente que permite recuperar archivos con extensión OWL desde repositorios poblados con anterioridad. Dicha tarea puede resultar increíblemente compleja, sobre todo si no se acota adecuadamente el espacio de trabajo en el cual la araña web operará.

Con la finalidad de hacer más eficiente el proceso, la araña web se dividió en dos principales componentes:

- Un módulo de raspado (scrape). Dicho módulo permite al araña visitar un directorio central con diferentes tipos de archivos. Dicho directorio se carga en memoria a través de una librería intrínseca de Java que permite enviar peticiones HTTP y capturar el resultado dentro de una variable. La tarea principal de este módulo es identificar todos los archivos con extensión OWL que estén presentes en el directorio central, sin importar el contexto semántico ni contenido adicional que pudiese estar presente también.

- Un analizador sintáctico. Este módulo permite la carga y el posterior análisis del contenido de un archivo OWL. Gracias a una interfaz de programación de aplicaciones (API) llamada OWL-API, no sólo es posible obtener el contenido general del archivo, sino que también se pueden almacenar partes específicas de la ontología, como por ejemplo: propiedades, clases, relaciones, reglas, etc.

El proceso utilizado para obtener los archivos de ontologías utilizó como base el buscador semántico Swoogle, desarrollado en la Universidad de Meryland, en Baltimore (UMBC).

Debido a las cotas establecidas en este proyecto terminal, se emitieron otro tipo de archivos relacionados a las ontologías, como son archivos RDF o incluso OWL2.

Sub-Módulo: Raspador (*scraper*)

La finalidad de este módulo es visitar remotamente una página determinada, a través de una URL específica. Esto se hará mediante una biblioteca incluida en el SDK de Java, la cual permite enviar peticiones HTTP desde una aplicación en Java, para posteriormente capturar la respuesta en una variable para su posterior manipulación.

De manera ilustrativa, a continuación se incluye una tabla en donde se puede ver el comportamiento general de este módulo.

Entrada	**Salida**
http://swoogle.umbc.edu/index.php?option=com_frontpafr&service=search&queryType=search_swd_ontology&sesearchStri=**palabra_clave**	Documento HTML con enlaces hacia archivos OWL
http://swoogle.umbc.edu	Documento HTML sin referencias a archivos OWL útiles en este proyecto terminal.

http://swoogle.umbc.edu/PaginaQueNoExiste	Código de error HTTP 404
http://swoogle.umbc.edu/robots.txt	Archivo de texto plano con directrices sobre políticas de acceso a carpetas y archivos

Si el módulo ha encontrado una página que contiene enlaces de hipertexto hacia archivos de extensión OWL, se inicia un proceso de análisis sintáctico del código HTML con la finalidad de eliminar etiquetas de elementos no deseados. El resultado de este análisis sintáctico permitirá la posterior extracción de las URLs de los archivos de ontologías.

En el siguiente paso, se hace una manipulación del código HTML. Primero se crea una lista con todos los enlaces de hipertexto presentes en la página, descartando cualquier otro elemento.

En segunda instancia, se hace un análisis de dichos enlaces para determinar si el recurso al cual enlazan se trata de una ontología. Esto se hace por medio de una comparación de cadena de caracteres, buscando la extensión OWL en la URL del recurso a analizar. Si se cumple este criterio de clasificación, se procede a la tercera etapa.

Finalmente, se hace una petición HTTP al recurso filtrado y se analiza la respuesta. Si se trata de una petición válida con una respuesta exitosa (código 200), se procede al almacenamiento local de la ontología en formato de archivo OWL. Utilizando el directorio local del proyecto como base, se crea una carpeta en donde se almacenarán todos los archivos con extensión OWL. Una vez que la escritura del archivo ha finalizado, se procede a cerrar la conexión del recurso y pasar al siguiente enlace de hipertexto, si es que hubiese uno.

Es importante es importante recordar que los tres pasos anteriores se realizan de manera interactiva, ya que es necesario realizar el mismo procedimiento en todos los enlaces de hipertexto que se encuentran en una URL predeterminada. Abordando el contexto técnico, la araña web hace un recorrido por amplitud (con sólo un nivel de profundidad).

Sub-Módulo: Análisis sintáctico (parser)

Este módulo comienza el análisis del código HTML de una URL específica. Dicho código contiene elementos no deseados y que se pretende filtrar de manera eficiente. Este filtrado se hace en varias capas, cada una de ellas eliminando elementos de cierto tipo.

La primera etapa contempla la eliminación de cualquier etiqueta HTML que no contenga un enlace de hipertexto a un recurso determinado. Es decir, etiquetas como <head>, <body>, <table>, etc. son eliminadas por completo, sin importar el contenido presente. Durante este análisis sintáctico no se toman en cuenta jerarquías ni la validez del código HTML. Asimismo se ignoran referencias a archivos invocados remotamente dentro del mismo código HTML, como pueden ser: definiciones de esquemas, hojas de estilo, código externo en otro lenguaje, archivos multimedia, etc.

A continuación se procede a eliminar cualquier enlace de hipertexto que haga referencia a cierto tipo de archivos y/o esquemas predeterminados. En las páginas de resultados del buscador Swoogle se puede encontrar una referencia a la licencia CreativeCommons, que enmarca las directrices de acceso a la información y a los mismos resultados de búsqueda. Dicho enlace contiene elementos XML que pudieran hacer creer a la araña web que se trata de un archivo de ontologías. Por esta situación, todos los enlaces que apunten hacia el dominio creativecommons.org son ignorados por el analizador sintáctico.

Si se ha llegado hasta esta etapa, con seguridad se puede afirmar que el recurso enlazado desde el código HTML se trata de un archivo con extensión OWL, que tentativamente contiene una definición válida de una ontología. Sólo queda almacenar este archivo de manera local para su posterior consulta de manipulación con la ayuda del módulo de búsqueda.

Cada vez que se encuentra una ontología, ésta se almacena con extensión OWL dentro de una carpeta específica ubicada en la carpeta del proyecto. Para evitar posibles colisiones (nombres de archivos repetidos), se genera un nuevo nombre de archivo en tiempo real utilizando una marca de tiempo (timestamp) como factor aleatorio. Asimismo se utiliza un número aleatorio (en el rango de 1 a 1000) como factor complementario para la generación del nombre de archivo que se va a almacenar de manera local.

Todos los nombres de archivo generados, carecen de un contexto semántico, ya que el módulo de búsqueda hará uso de ellos posteriormente.

Módulo: buscador Web

Una vez que se han obtenido diferentes archivos con extensión OWL, es trabajo del módulo de búsqueda, el brindar una interfaz para que los usuarios puedan ingresar palabras clave con el objetivo de identificar todas las ontologías que coincidan con su criterio de búsqueda, es decir, se mostrarán todas las ontologías que incluyan al menos una vez el término ingresado.

Dicho motor de búsqueda realiza una búsqueda iterativa a través de todos los archivos de ontologías almacenados de forma local. Es decir se hace una comparación archivo por archivo para comprobar si la palabra clave ingresada por usuario está presente. Este método de acceso es fácil de implementar y no requiere de una conexión a Internet, por lo que resulta una solución práctica. Sin embargo, la escalabilidad es un factor en detrimento de esta solución. Si se desease hacer un índice con miles de archivos, el desempeño del buscador sería notablemente menor ya que existiría una limitante en la velocidad de escritura/lectura de archivos.

Debido a las limitaciones establecidas en este proyecto terminal, se decidió realizar la solución basada en archivos ya que los índices generados no son muy grandes.

Sub-Módulo: Cálculo del factor de relevancia

El factor de relevancia es un número flotante que permite saber el grado de relevancia de una determinada ontología con respecto a una palabra clave ingresada por el usuario. Se trata de un cociente que permite identificar el rango de relevancia de un archivo OWL tomando en cuenta la frecuencia de aparición de una palabra clave dentro de las clases de una ontología.

El cálculo de dicho factor de relevancia proporciona una forma fácil y eficaz de medir en primera instancia la relevancia de un archivo en el contexto de una o varias palabras clave.

Para calcular el factor de relevancia, se utilizó la siguiente fórmula:

$$f = \frac{|D_1|}{|C_1|}$$

Para poder realizar el cálculo, es necesario generar dos conjuntos a partir del contenido de la ontología y de la palabra clave ingresada por el usuario.

En primer lugar se obtiene el conjunto de todas las clases pertenecientes a la ontología. Esto se hace con ayuda de la OWL-API que provee de una función específicamente para este propósito. A este conjunto, se le añadirán todas las palabras clave ingresadas por el usuario. Posteriormente, se obtendrá la *cardinalidad* del conjunto resultante, denominado *C1* para este ejemplo.

Asimismo se necesita obtener el conjunto denominado *D1*, para efectos de este ejemplo. Dicho conjunto contendrá todos los elementos presentes tanto en el conjunto de clases de la ontología, así como los elementos presentes obtenidos a partir de la palabra clave (o palabras clave) ingresadas por el usuario. De igual manera se obtendrá el tamaño de este conjunto (*cardinalidad*), ya que será usada en la siguiente etapa.

Finalmente se hará una división entre la cardinalidades de los conjuntos mencionados anteriormente. El cociente es conocido como factor de relevancia.

Sub-Módulo: Analizador de relevancia

Tan pronto como el usuario ha ingresado una palabra clave (o varias), es necesario obtener las ontologías relevantes que se van a mostrar. Para esto, es necesario hacer una búsqueda de forma iterativa, para calcular el factor de relevancia de cada una de las ontologías presentes en el directorio local.

Antes de mostrar los resultados de búsqueda al usuario, es necesario definir un umbral (threshold) de relevancia. Todas las antologías cuyo factor de relevancia estén por encima del umbral serán mostradas, la restantes simplemente serán ignoradas y el usuario no sabrá que fueron consultadas. Entre más pequeño sea el umbral, habrán más resultados de búsqueda, aunque la relevancia será menor.

Con el umbral de relevancia definido, el primer paso es acceder al directorio local en donde se almacenan los archivos con extensión OWL. Mediante un ciclo, se lee cada uno de los archivos presentes y se realiza el cálculo del factor de relevancia para cada

ontología. Una vez que se ha obtenido el factor, se compara con el umbral predefinido, y en caso de cumplir los requerimientos antes explicados, la ontología será mostrada al usuario para su consulta.

Dependiendo del número de archivos presentes, el proceso del análisis de relevancia puede tomar desde algunos milisegundos hasta varios minutos.

Implementación

Computadora de escritorio

- Procesador Intel® Core™ i7 @ 3.4 GHz.
- 16 GB de memoria RAM @ 1200 MHz.
- Disco duro 320 GB / 7200 rpm.
- Sistema Operativo Mac OS X Mountain Lion 10.8.2.

Servidor remoto

- Procesador Intel® Xeon™ v1234 @ 3.4 GHz.
- 32 GB de memoria RAM 1200 MHz.
- Disco duro 2x(1 TB), RAID-0.
- Sistema operativo Windows Server 2008 R2 SP1.

Software, APIs y otras herramientas

- Para la correcta elaboración y funcionamiento de este proyecto terminal, se habrá de adoptar un entorno de desarrollo específico.

- En primer lugar, se usará el lenguaje de programación Java. Para tal fin, se empleará el paquete de desarrollo Java Platform (JDK) 7u17.

- Se usará el servidor web Apache Tomcat v 7.0.34.

- El proceso de análisis y manejo de ontologías se hará mediante el uso de algunas bibliotecas externas. Principalmente, se hará uso de OWL API versión 2.0 para la creación, manipulación y serialización de ontologías en formato OWL. También se usará el marco de trabajo PrimeFaces versión 4.0, publicado bajo licencia Apache.

- Finalmente, y como herramienta auxiliar, se usará el editor de ontologías Protégé para crear representaciones visuales y modelos a partir de ontologías. Soporta los formatos OWL, RDF e incluso XML. Se usará Protégé versión 4.1 (distribución estable) para el sistema operativo Mac OS X.

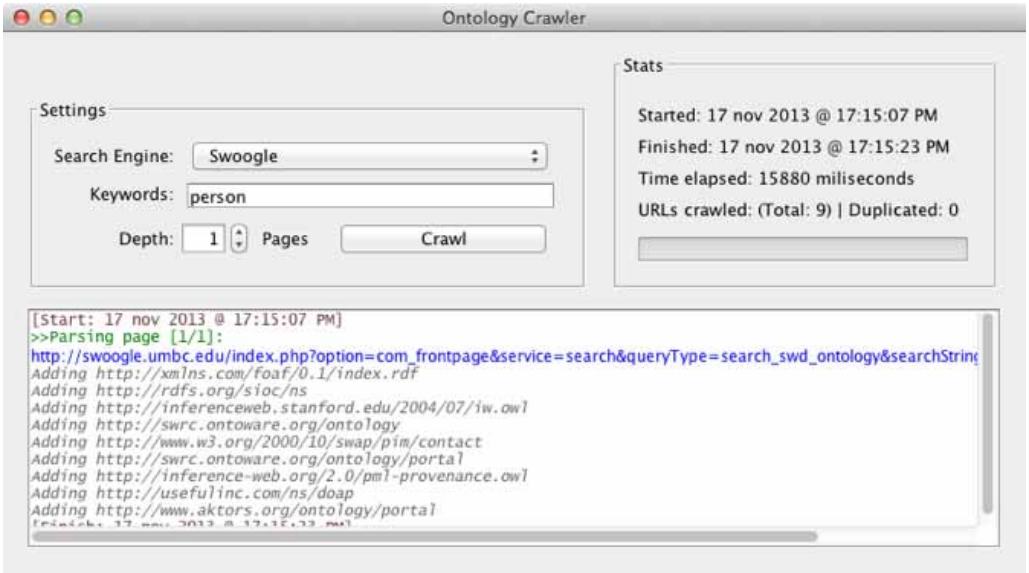
- Esta aplicación será capaz de manejar a lo sumo 50 archivos de ontologías de manera simultánea. Los archivos antes mencionados tienen formato OWL o RDF

Pruebas del sistema

Una vez se concluyó la implementación del sistema, fue necesario elaborar pruebas para asegurarse del correcto funcionamiento de todos los módulos. Se realizaron pruebas con diferentes tipos de parámetros y bajo diferentes condiciones de ejecución.

Pruebas de la araña Web

La primera prueba realizada fue una simple recuperación de ontologías, usando como palabra clave, el término *person*.



Mediante el archivo de registro (*log*) se puede ver la actividad realizada por la araña Web. Por cada página explorada, se muestran las ontologías leídas correctamente y que se almacenan de manera local.

Pruebas del módulo de búsqueda (consola)

Como primera prueba, se introdujo el término *person* como palabra clave. El buscador analizó todos los archivos OWL almacenados localmente y mostró las coincidencias apropiadas en forma de lista. Se usó un umbral de 0.01. En cada resultado se puede ver también una lista de términos que coinciden con la(s) palabra(s) clave:

```
run:
Enter your query here:
person

Threshold: 0.01
Query: person

Relevance: 0.06666667
Matches: person
1.- [File]: 10761384730117.owl
========================================

Relevance: 0.083333336
Matches: person
2.- [File]: 13601373446452.owl
========================================

Relevance: 0.03448276
Matches: person
3.- [File]: 13741373447016.owl
========================================
```

Se repitió la misma prueba, esta **vez** usando un umbral mucho mayor (0.95). No se mostraron resultados que coincidieran con los parámetros definidos, ya que ninguna ontología tiene un factor de relevancia mayor o igual a 0.95

```
run:
Enter your query here:
person

Threshold: 0.95
Query: person

No results to show
```

Como tercera prueba se realizó una consulta con un umbral de 0.01, usando el término *t-rex*.

```
run:
Enter your query here:
t-rex

Threshold: 0.01
Query: t-rex

No results to show
```

En el directorio local no se hallaron ontologías que tuvieran alguna coincidencia con dicho término, por lo que se le hace saber al usuario que no hubieron coincidencias.

Como prueba final, se usaron varias palabras clave (separadas por espacios). Para este ejemplo, se usó *person animal* como entrada, y se obtuvieron todaslas ontologías con coincidencias parciales:

```
run:
Enter your query here:
person animal

Threshold: 0.01
Query: person animal

Relevance: 0.0625
Matches: person
1.- [File]: 10761384730117.owl
========================================

Relevance: 0.07692308
Matches: person
2.- [File]: 13601373446452.owl
========================================

Relevance: 0.2857143
Matches: person, animal
11.- [File]: 42041373446632.owl
========================================
```

Prueba del buscador web: Se realizó una búsqueda usando *person* como palabra clave y con un umbral de 0.05. Estos fueron los resultados:

#	File	Relevancy
1	13601373446452.owl	0.083
2	13741373447016.owl	0.034
3	17361373446582.owl	0.167
4	28251373446950.owl	0.033
5	29551373446952.owl	0.013
6	31391373446714.owl	0.023
7	36441373446308.owl	0.067
8	39161373446306.owl	0.03
9	39931373446716.owl	0.2
10	42041373446632.owl	0.143
11	42931373446951.owl	0.013
12	46531373446661.owl	0.043

Bibliografía

[1] W. Lacy, *Owl: Representing Information Using The Web Ontology Language*. Bloomington: Trafford Publishing, 2005, pp. 4-5.

[2] L. Yu, *A Developer's Guide to the Semantic Web*. Manhattan:Springer Publishing, 2011.

[3] E. Sánchez, *Clasificación de servicios Web semánticos mediante ontologías, propuesta de proyecto terminal, Universidad Autónoma Metropolitana,* México D.F., México, 2012.

[4] J. P. Martínez, *Extracción automatizada y representación de servicios Web mediante ontologías, propuesta de proyecto terminal, Universidad Autónoma Metropolitana, México D.F., México, 2011*

[5] L. Y. García, *Sistema de recuperación de información de textos de investigación de la Web, propuesta de proyecto terminal, Universidad Autónoma Metropolitana, México D.F., México, 2012*.

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Sistema de búsqueda y recuperación de ontologías

Manual de Usuario

Alan Christopher Alamillo Medina

Matrícula: 208333483

Asesores

Dra. Maricela Claudia Bravo Contreras

Profesor Asociado

Departamento de Sistemas

Dr. José Guadalupe Rodríguez García

Departamento de Computación del CINVESTAV

Unidad Zacatenco

Noviembre 2013

Tabla de contenido

Requisitos del Sistema 3
Lanzamiento de las aplicaciones 3
Interfaz Gráfica 3
Araña Web 3
Validación de campos 4
Buscador Web 5

Requisitos del Sistema

Para poder ejecutar el sistema de manera correcta, es necesario contar con los siguientes puntos:

- Sistema Operativo capaz de ejecutar el *Java Development Kit*.
- Tener el Java Development Kit (JDK) instalado. Se puede descargar desde http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Tener una conexión a Internet.
- Servidor Web Apache Tomcat corriendo en el puerto 8080. Tomcat se puede descargar desde http://tomcat.apache.org/download-70.cgi.

Lanzamiento de las aplicaciones

Las aplicaciones se encuentran compiladas en archivos con extensión .jar. Basta con hacer doble clic sobre el archivo .jar correspondiente para que el sistema operativo ejecute la aplicación. Para este proyecto, hay dos archivos ejecutables:

- OntologyCrawler.jar (ubicado en la carpeta dist, dentro de la carpeta del proyecto OntologyCrawler)
- OntologySearchEngineWeb.war (ubicado en la carpeta dist, dentro de la carpeta del proyecto OntologySearchEngineWeb). Para poder ejecutar esta aplicación correctamente, se necesita tener activo el servidor Web Apache Tomcat en el puerto por defecto (8080).

Interfaz Gráfica

Araña Web

Una vez que se ha ejecutado la aplicación OntologyCrawler.jar, el usuario verá una ventana con los controles principales de la araña web (Figura 1) .

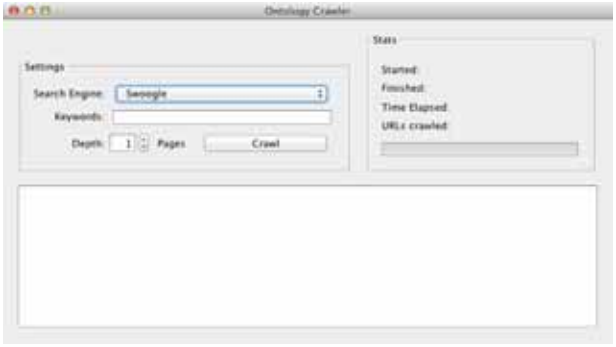


Figura 1. La interfaz principal de la araña Web.

La interfaz requiere de 3 campos de entrada para funcionar.

- **Search Engine (Motor de búsqueda):** En esta lista se incluyen los motores de búsqueda de ontologías desde los cuales se hace la recuperación de archivos OWL. Por el momento, solo *Swoogle* es soportado por la araña web.

- **Keywords (Palabras clave):** Para poder encontrar ontologías es necesario usar palabras clave para generar un filtro. Se pueden usar términos de una sola palabra, o bien, si se requiere usar más de una palabra, éstas deberán estar separadas por un espacio. Entre más palabras se usen, menos resultados se generarán.

- **Depth (Profundidad):** Este valor indica cuántas páginas de resultados de Swoogle se deben analizar en busca de archivos OWL. Entre más grande sea este valor, probablemente haya más resultados, pero el tiempo empleado será mucho mayor.

Cuando se han establecido los valores en los tres campos mencionados antes, solo resta hacer clic en el botón con la leyenda *"Crawl"*. Esto dará inicio al proceso de exploración y recuperación de ontologías. El proceso se puede ver en el área de registro (log) del programa (Figura 2).

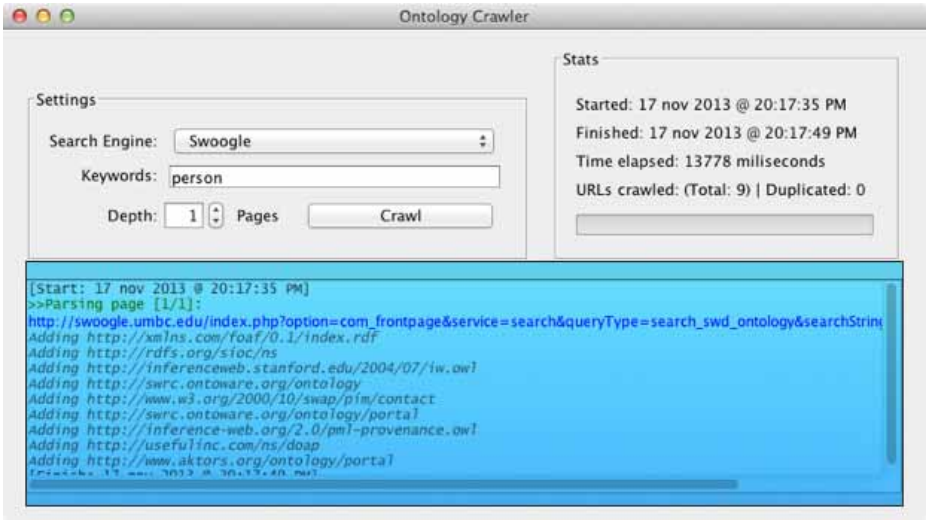


Figura 2. El área de registro de la araña Web resaltada dentro de un rectángulo.

Validación de campos

Existen dos reglas de validación presentes en la ventana de la araña web. En caso de incurrir en algún tipo de error, un mensaje mostrará más detalles al respecto (Figura 3).

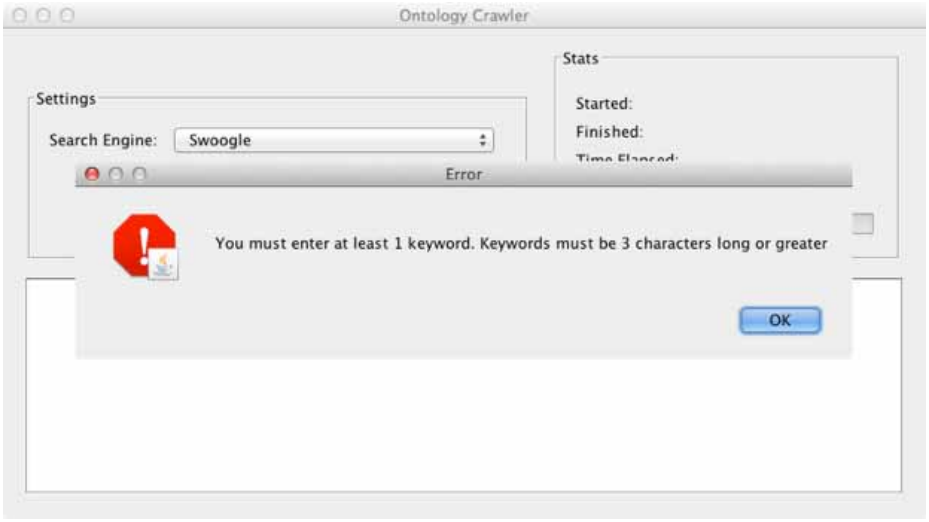


Figura 3. Ejemplo de validación de datos de entrada en acción

- **Palabras clave:** Debe haber por lo menos una palabra clave, y ésta debe ser de al menos tres caracteres de longitud. Se aceptan símbolos especiales, aunque esto repercutirá en el número de resultados encontrados.

- **Profundidad:** Debe ser un número entero mayor o igual a 1. Aunque no existe una limitación explícita sobre el número de páginas que se pueden analizar, se recomienda usar valores no mayores a 10, puesto que después de la página 10, los resultados o bien no son relevantes, o se trata de ontologías procesadas previamente.

Buscador Web

Una vez que se esté ejecutando el servidor Web Apache Tomcat, el usuario será capaz de visualizar la interfaz del buscador en la siguiente URL: http://localhost:8080/OntologySearchEngineWeb/

El usuario visualizará entonces una ventana similar a la que se puede ver en la Figura 4.

Figura 4. La interfaz principal del buscador web

En la ventana principal hay un formulario con dos campos:

Keywords: En este campo se introducirán las palabras clave que se usarán para buscar.

Threshold: Es el umbral de relevancia. Debe ser un número entre 0 y 1, donde 0 representa una relevancia nula y 1 una relevancia total. Entre más cercano esté el valor a 1, menos resultados habrá, pero los que existan, serán más relevantes.

Con los campos llenos de acuerdo al criterio del usuario, solo resta dar clic en el botón Search. Una vez que el motor de búsqueda arroje los resultados pertinentes, estos se encontrarán desplegados de forma similar a lo que se ve en la Figura 5.

#	File	Relevancy
1	13601373446452.owl	0.083
2	13741373447016.owl	0.034
3	17361373446582.owl	0.167
4	28251373446950.owl	0.033
5	29551373446952.owl	0.013
6	31391373446714.owl	0.023
7	36441373446308.owl	0.067
8	39161373446306.owl	0.03
9	39931373446716.owl	0.2
10	42041373446632.owl	0.143
11	42931373446951.owl	0.013
12	46531373446661.owl	0.043

Figura 5. Página de resultados de búsqueda.