OntoLing Annotizer

Una herramienta de ayuda para la anotación lingüística

Proyecto de Sistemas Informáticos

Facultad de Informática

UNIVERSIDAD COMPLUTENSE
MADRID

Autor: Martín Montalvo Martínez

Profesor director: Antonio Pareja Lora

Curso: 2008/2009

AUTORIZACIÓN DE USO

Por el presente documento autorizo a la Universidad Complutense y a su director, D. Antonio Pareja Lora, a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Firma del autor:

Martín Montalvo Martínez

RESUMEN

El objetivo del presente proyecto es desarrollar una herramienta (OntoLing Annotizer) destinada a la anotación semántica de documentos. La anotación semántica de documentos, tiene un papel principal dentro de la Web Semántica, pues explicita el significado o la semántica de los recursos de la web, para que este pueda ser “entendido” por los computadores. OntoLing Annotizer explicita la semántica de los documentos mediante anotaciones, las cuales son añadidas en un documento distinto del original, teniendo así el texto y las anotaciones separadas. El sistema de anotación se sustenta en cuatro tipos de ontologías lingüísticas: la de unidades, la de atributos, la de valores y la de relaciones. Cada una de las anotaciones pertenece a un concepto de la ontología de unidades. Estas unidades pueden tener unos atributos y estos atributos toman unos valores. Finalmente, las unidades (y por tanto, las anotaciones) pueden relacionarse unas con otras.

ABSTRACT

This project aims at developing a tool for the semantic annotation of documents, called "OntoLing Annotizer”. The semantic annotation of documents has an important role in the Semantic Web, since it makes explicit the meaning or the semantics of web resources. Thus, it can be ‘understood’ by computers. OntoLing Annotizer makes the meaning of documents explicit by means of annotations, which are included into a different document from the original one, keeping the original text and its annotations separated. The annotation process is based on four different types of ontologies, related to linguistic units, attributes, values and relationships respectively. Each of the annotations belongs to a concept of the ontology of units. These units may have some attributes and these attributes can take certain values. Finally, units (and, hence, annotations) can be interrelated as well.

ÍNDICE

INTRODUCCIÓN

En estos días la Web es un gigantesco almacén virtual, donde se encuentra una cantidad ingente de información, accesible para todo el mundo y donde todo el que quiera puede depositar el contenido que desee, siempre que posea un computador. La web actual o World Wide Web está enfocada al acceso y/o intercambio de información entre personas, aunque la información se envía a través de máquinas, y esta información solo es comprensible y asimilada por las personas. Aunque esto parece lógico, ya que el fin último es este, también sería positivo que la información fuese comprensible por las máquinas, y que estas fuesen capaces de procesarla atendiendo a su contenido y no solamente enviarla y formatearla para una correcta visualización por las personas.

Al principio, la Web tenía un reducido número de documentos, los cuales eran muy rudimentarios y realizados en su totalidad por personas, sin una estructura definida. Todos estos documentos están construidos mediante metalenguajes basados en etiquetas que proporcionan información para que el navegador, con el que se accede a los documentos presentes en la Web, realice una correcta presentación de los contenidos del documento. La finalidad de las etiquetas de estos lenguajes es principalmente estética, ya que no contemplan ningún aspecto que tenga que ver con la semántica o el significado.

Con el pasar del tiempo, el número de documentos presentes en la Web ha crecido de forma exacerbada, de forma que procesar esta información sin la ayuda del computador se torna imposible. Los motores de búsqueda actuales obtienen sus resultados consultando una base de datos, creada por un programa, indexando los contenidos a partir de algunos datos de estos. Según Vossen (2007), la técnica utilizada por uno de los buscadores más importantes y notables, Google (www.google.es), se basa en el algoritmo

PageRank. El algoritmo PageRank, explicado de forma simple, se basa en la existencia de las palabras introducidas como entrada en las páginas que forman parte de su base de datos. Lo novedoso de este algoritmo fue que ordenaba los resultados según su puntuación, la cual venía dada por el número de referencias de otras páginas a la que se asigna la puntuación. Pero el valor de esta puntuación depende de la credibilidad que posea esta página que la referencia, y se calcula de forma recursiva.

Aunque muchos de los documentos de la web convencional ya contienen etiquetas en XML, estas etiquetas no le permiten al computador tener un conocimiento semántico, y el contenido sigue siendo incomprensible para el computador. Que el computador entienda el significado del texto es el principio básico de la Web Semántica. Con la Web Semántica se intenta que los resultados mostrados por los buscadores sean más exactos y, por ende, que se obtenga un menor número de páginas (más preciso). Al disminuir el número de páginas que propone el buscador, y estar estás más cerca del resultado esperado, se reduce el tiempo y el esfuerzo de las personas a la hora de filtrar la información. Para conseguir el anterior objetivo, las páginas (constituidas de texto, imágenes, sonidos, y cualquier otro tipo de material) deben estar anotadas. Estas anotaciones suelen estar en documentos aparte, pero vinculadas de alguna forma, o embebidas dentro de los documentos que contienen la información usando algún metalenguaje de etiquetas. La mayoría de estos metalenguajes están basados en RDF (http://www.w3.org/TR/rdf-primer/) y descritos mediante sintaxis XML (http://www.w3.org/XML/).

Al proceso de agregar estas anotaciones o metadatos semánticos, con los que el computador comprenderá la semántica (o el significado) del documento, se le denomina anotación semántica (Castellanos, 2007). Para realizar estas anotaciones, existen unas aplicaciones llamadas herramientas de anotación semántica. Con el fin de que el resultado obtenido proporcione

una información que sea fácil de recuperar y procesar, se requiere de un diseño avanzado y de la aplicación de modelos y formalismos. Si bien recientemente los expertos en desarrollo de ontologías, han utilizado estas para la anotación semántica (Benjamins et al., 1999; Motta et al., 1999; Luke et al., 2000; Staab et al., 2000), también se debería tener en cuenta los resultados sobre anotaciones obtenidos en el campo de la Lingüística de Corpus, no sólo en el nivel semántico, sino también en el resto de niveles lingüísticos (Aguado, 2002a).

Dentro de este panorama, la aplicación que se ha desarrollado, OntoLing Annotizer, se sitúa dentro de las herramientas de anotación semántica. La herramienta generada es capaz de anotar documentos en formato HTML por medio de ontologías. Estas ontologías deben seguir un modelo concreto, basado en cuatro tipos diferentes de ontologías. En concreto, estos cuatro tipos de ontologías, cada uno con un cometido distinto, son: las ontologías de unidades, las ontologías de atributos, las ontologías de valores y las ontologías de relaciones.

Cada una de las anotaciones que se realicen sobre el documento, aunque estén almacenadas en otro archivo, referenciará un concepto perteneciente a la ontología de unidades. De esta forma, todas las anotaciones simbolizan algún concepto de la ontología de unidades (en inglés, *units*). En otra ontología, la ontología de atributos (*attributtes*), están presentes todos los atributos que pueden tener las anotaciones y, a su vez, estos atributos tomarán unos valores. Todos los valores que puedan tomar los atributos también se encuentran en una ontología. Esta ontología es la de valores (*values*). Como las unidades también se relacionan con otras unidades, existe una última ontología, cuyos conceptos son los tipos de relaciones que pueden darse entre unidades, y que se denomina ontología de relaciones (*relationships*).

A continuación se incluye (en el capitulo denominado Estado del arte) el estudio preliminar que hubo de llevarse a cabo antes de abordar la construcción de la herramienta OntoLing Annotizer. Posteriormente, se presentarán los objetivos de la misma, en el capítulo Objetivos del proyecto. En el capitulo Desarrollo del proyecto se detallan todos los principales pasos seguidos para llegar a construir la herramienta, exponiendo el problema e inmediatamente como se solucionó. Para el capítulo Conclusiones, se explican las conclusiones a las que he llegado, tanto sobre el contexto en el que se encuadra la herramienta, como sobre la propia herramienta. En el capítulo siguiente, Trabajos futuros, se sugieren futuras líneas de trabajo, y así poder ampliar la herramienta. Los dos siguientes capítulos son las referencias bibliográficas, y direcciones de internet que se han utilizado y/o mencionado. Por último, está el Apéndice: Manual de usuario, que especifica: los requisitos necesarios para ejecutar la herramienta; que pasos seguir para lanzar la aplicación; y una lista con las principales funciones, explicando con gran detalle los pasos a seguir en cada una de estas funciones.

ESTADO DEL ARTE

En este apartado se procederá al estudio del estado del arte, es decir, se realizará un recorrido por el contexto actual en el que se basa el trabajo. Se repasarán las principales tecnologías empleadas, así como un análisis de las herramientas actuales.

TECNOLOGÍAS EMPLEADAS

En este punto se abordarán las tecnologías relacionadas de alguna forma con la herramienta, ya sea de forma directa o con una implicación indirecta, pero que requieren de una explicación previa para poder tener una panorámica general.

WEB SEMÁNTICA

La Web Semántica es una extensión de la actual Web, dotada de un mayor significado. Esta nueva extensión provee un marco común que permite que los datos sean compartidos y reutilizados a través de la aplicación, la empresa, y las fronteras de las comunidades. Se trata de un esfuerzo colaborativo liderado por el W3C (http://www.w3.org/), con la participación de un gran número de investigadores y socios industriales. Se basa, principalmente en el uso del lenguaje RDF (Resource Description Framework, http://www.w3.org/RDF/).

En la Web actual, conocida como World Wide Web, la información sólo es comprensible para los humanos: se facilita su comprensión añadiendo imágenes, sonido y gráficas; posicionando el texto de una forma agradable a la vista humana; algunos sitios web tienen traducciones del lugar para

distintos idiomas; existen opciones de accesibilidad para personas discapacitadas; etc. Resumiendo, todo el contenido está orientado al consumo humano. Todas estas características son totalmente válidas, y bajo mi punto de vista deberían conservarse, y ser compatibles con las características de la Web Semántica. Sin embargo, el aspecto más destacable de esta nueva Web sería que el contenido no será sólo comprensible por los humanos, sino que también lo será por la máquinas. La comprensión por parte de las máquinas se refiere a la semántica (significado) de la información, que estas podrán "entender y procesar": qué explica un texto, qué se visualiza en una imagen, qué sucede si se ejecuta una aplicación web, qué pasos dar para conseguir un objetivo, etc.

Para conseguir que el contenido sea comprensible por parte de los computadores, se deben adjuntar una serie de etiquetas en un metalenguaje, llamadas anotaciones. El proceso por el cual se dota a los documentos de una semántica comprensible por los computadores se denomina anotación semántica, y se tratará en el siguiente apartado. Las anotaciones son metadatos que pueden incluirse en el mismo documento o en un documento aparte, pero vinculadas de alguna forma. Los metadatos son datos que se refieren a los propios datos, datos de datos. La doble visión de los metadatos, como datos o metadatos, depende de la perspectiva. Desde el punto de vista del texto, son metadatos, ya que proporcionan información sobre los datos, pero para un programa de la Web Semántica son sus datos de entrada, lo que debe procesar.

Si se quiere que la web sea comprensible por las máquinas, debería serlo por todas y en cualquier sitio, por lo que parece lógico que no haya problemas con los idiomas. Así que el conocimiento debería ser expresado de forma universal y de mutuo acuerdo. En este punto juegan un papel destacado las ontologías, como forma de explicitar el conocimiento de forma consensuada.

Estando ya implementada la Web Semántica se podrá disponer de agentes software, conocidos en los entornos de la Inteligencia Artificial (IA) como agentes inteligentes. No existe ninguna definición de agente que haya sido plenamente aceptada por la comunidad científica, pero una de las más sencillas puede ser la que lo considera un ente que percibe el entorno en el que se encuentra y actúa en consecuencia (Russel, 2004). Un agente es capaz de recoger la información de la red, procesarla e interactuar con otros agentes y con el mismo usuario. Estos agentes tendrían una autonomía casi completa, que en algunas tareas podría ser total. En un principio, el agente tendría un margen de decisión muy limitado, ya que el conocimiento a priori es reducido, y este irá incrementándose con el paso del tiempo, gracias a la experiencia. Esta experiencia podría adquirirse de dos formas: interactuando con el usuario mediante formularios o, de forma pasiva, observando las acciones del usuario. Un ejemplo muy ilustrativo, con el que se observa la utilidad de los agentes, sería cómo son capaces de encontrar una clínica que cubra el seguro médico, próxima al domicilio, y concertar posteriormente una cita, de acuerdo con la agenda del paciente (Berners-Lee, 2001).

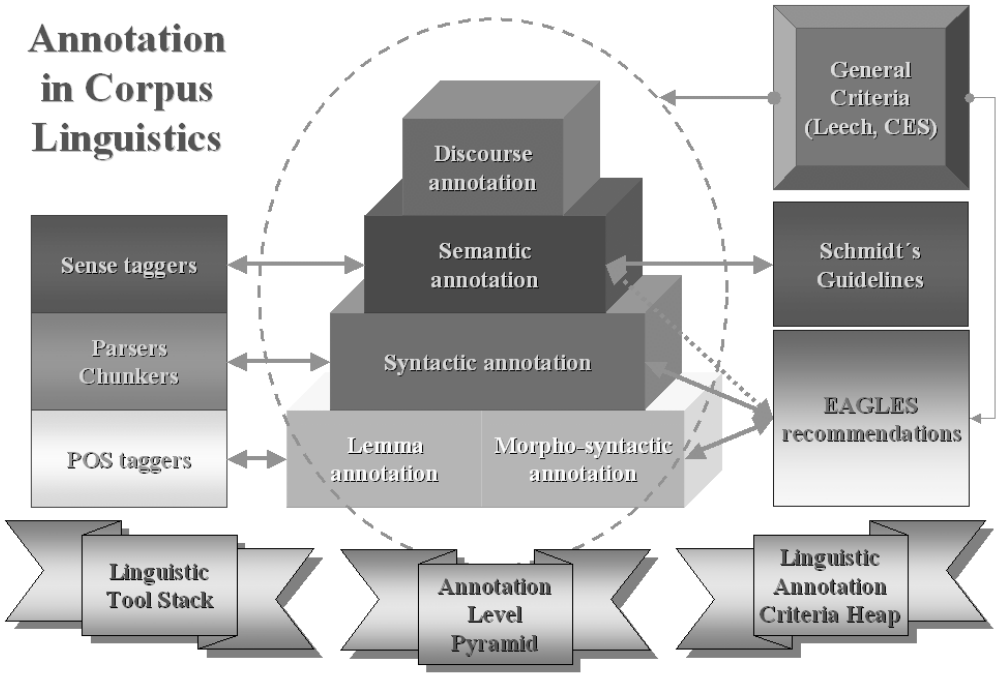
ANOTACIÓN SEMÁNTICA

Como se introdujo en el apartado anterior, mediante la anotación semántica de los documentos de la web tradicional, se consigue llegar a la Web Semántica. Para que los computadores lleguen a comprender la información contenida en páginas web y documentos, se debe añadir metadatos semánticos a estos contenidos. Los metadatos constituyen las anotaciones que se crean durante el proceso de anotado. Cada uno de los recursos web (páginas web, documentos de texto, imágenes, sonido, video, etc.) anotados necesita de un identificador único, capaz de distinguirlo, de forma que se le puedan asociar características, relaciones, aserciones lógicas o cualquier tipo de metainformación. Esta necesidad es uno de los motivos

por los que se utiliza el lenguaje XML para crear las anotaciones semánticas, ya que este lenguaje hace uso de los URIS, *Uniform Resource Identifiers* (en español, Identificadores de Recursos Uniformes) y los XML Namespaces (espacios de nombres de XML, http://www.w3.org/TR/REC-xml-names/#sec-namespaces).

Por ser el pilar fundamental de la Web Semántica, la anotación semántica tiene una gran importancia dentro de esta. Se han estudiado y se siguen estudiando distintos modelos y formas de anotación semántica. Entre estos modelos se encuentran el de la Inteligencia Artificial (Benjamins et al., 1999; Motta et al., 1999; Luke et al., 2000; Staab et al., 2000), basado en el uso de ontologías, y los prototipos del campo de la Lingüística de Corpus (Aguado 2002a). Estos modelos no son incompatibles, sino que existe un modelo híbrido, en el que se toman los criterios y elementos tanto de la anotación ontológica como de la lingüística (Aguado 2004). Dentro de la anotación lingüística existen varios niveles de anotación, a saber: anotación de lemas, anotación morfosintáctica, anotación sintáctica, anotación semántica y anotación del discurso. En la **Figura 1**, se muestran estos niveles, junto a sus correspondientes herramientas (Linguistic Tool Stack) y los criterios, recomendaciones y directrices aplicables (Linguistic Annotation Criteria Heap). Según Aguado (2002b), el significado de los documentos no está presente únicamente en el nivel semántico, sino que está implícito y distribuido por todos los niveles. Así que no sólo hay que realizar anotaciones en el nivel semántico, sino que hay que realizar un anotado multinivel, abarcando los cinco niveles, para hacer explícito por completo dicho significado.

Figura 1: Niveles, herramientas y recomendaciones para la anotación lingüística de textos (tomado de Aguado et al 2004).



ONTOLOGÍAS

Las definiciones de ontología como esquema conceptual derivan de la filosofía. Actualmente no existe ninguna definición de ontología aceptada por toda la comunidad científica, pero lo que si existe son varias definiciones de esta que están más o menos extendidas y reconocidas. Una de las primeras y más extendidas es la de Gruber (1993):

> *"Una ontología es una especificación explícita de una conceptualización. El término proviene de la filosofía, donde una ontología es un recuento sistemático de la existencia. En sistemas de Inteligencia Artificial, lo que existe es lo que puede ser representado. Cuando el conocimiento de un dominio se representa mediante un formalismo declarativo, el conjunto de objetos que puede ser representado se llama universo del discurso. Esos conjuntos de objetos, y las relaciones que se establecen entre ellos, son reflejados en un vocabulario con el cual representamos el conocimiento en un sistema basado en conocimiento. Así, en el contexto de la*

Inteligencia Artificial, podemos describir la ontología de un programa como un conjunto de términos. En tal ontología, las definiciones asocian nombres de entidades del universo del discurso con textos comprensibles por los humanos que describen el significado de los nombres, y axiomas formales que limitan la interpretación y el buen uso de dichos términos. Formalmente, una ontología es una teoría lógica"

Esta definición no ha convencido a muchos autores y han surgido otras diferentes. En este estudio nos quedaremos con la de Borst (1997) por ser de las más cortas y concisas. Esta establece una ontología como *"una especificación formal (y explícita) de una conceptualización compartida"*. Para esclarecer esta definición se hará uso de la descomposición en conceptos, y su posterior descripción, efectuada por Pareja (2007).

- Por ser una **descripción**: está formada por conceptos, propiedades, relaciones, funciones, restricciones (reglas) y axiomas.
- Por ser **formal**: es computable.
- Por ser **explícita**: el tipo de conceptos utilizados, así como sus restricciones de uso, son expresados de forma clara y precisa.
- Por ser una **conceptualización**: es un modelo abstracto de un cierto fenómeno real, que identifica sus componentes (conceptos) más importantes.
- Por ser **compartida**: plasma un conocimiento consensuado.

En el caso de la Web Semántica, se utiliza para representar el conocimiento asociado a la semántica (el significado) del recurso, para que posteriormente pueda ser recuperado por un motor de búsqueda semántico o un agente inteligente. Este último hará uso de inteligencia artificial para realizar deducciones y para desempeñar las tareas que se le encomienden.

Los componentes principales de las ontologías varían según con que técnicas de representación y razonamiento se describan. Por ejemplo, si se utilizan marcos de Minsky y lógica de primer orden tenemos cinco tipos de elementos, a saber: las clases, que representan conceptos del dominio que se modela; las relaciones, que son las correspondencias entre conceptos de ese

dominio; las funciones, que son una caso especial de las relaciones, donde el n-ésimo elemento es único una vez fijados los n-1 anteriores; los axiomas formales, que modelan los enunciados que son siempre ciertos, y representan conocimiento que no puede ser definido formalmente por ninguno de los otros componentes; y por último las instancias, que son elementos o individuos de la ontología. Pero si se utiliza lógica descriptiva se tienen tres tipos de componentes, a saber: los conceptos, que representan clases de objetos (son equivalentes a las clases en el paradigma basado en marcos); los roles, que describen relaciones binarias entre conceptos y propiedades de los conceptos; y finalmente los individuos, que son las representaciones de las instancias de los conceptos (clases) y los valores que adquieren en sus roles (propiedades).

Al no tener una definición definitiva y comúnmente aceptada, tampoco se está en posición de hacer una clasificación única de ontologías. Pero, como refleja Castellanos (2007) se suelen seguir tres criterios: el tipo de conocimiento que contienen, la motivación de la ontología y el grado de formalismo. A su vez, dentro de cada criterio existen numerosas clasificaciones, así que aquí se mostrarán algunas de las más importantes.

Atendiendo al criterio de **tipo de conocimiento** contenido se muestran dos clasificaciones. Según Heijst *et al.* (1997) existen tres tipos de ontologías:

- **Ontologías terminológicas, lingüísticas**. Especifican los términos usados para representar conocimiento en un dominio determinado.
- **Ontologías de información**. Especifican la estructura de los registros de la base de datos.
- **Ontologías para modelar conocimiento**. Especifican conceptualizaciones de áreas del conocimiento.

Otra clasificación posible es la que se muestra a continuación, de Mizoguchi et al (1995):

- **Ontologías del dominio**. Contienen todos los conceptos asociados a un dominio particular.
- **Ontologías de tareas**. Describen el modo de utilizar el conocimiento del dominio para realizar sus tareas específicas.
- **Ontologías generales**. Contienen descripciones generales sobre objetos, eventos, relaciones temporales, relaciones causales, modelos de comportamiento y funciones.

Atendiendo al criterio de la **motivación** por la que se crea la ontología se muestran dos clasificaciones. Según la primera clasificación se tienen tres tipos de ontologías (Gómez Pérez *et al.*, 2003):

- **Ontologías para la representación de conocimiento**. Permiten explicar las conceptualizaciones que subyacen en los formalismos de representación de conocimiento.
- **Ontologías genéricas.** Definen conceptos comunes a diferentes áreas.
- **Ontologías del dominio.** Definen conceptualizaciones específicas del dominio.
- **Ontologías de aplicación.** Están ligadas al desarrollo de una aplicación concreta.

La siguiente clasificación la propuso Poli (2000) y en ella se consideran seis categorías:

- **Ontologías generales.** Tienen que ver con las categorías fundamentales y sus conexiones de dependencia.
- **Ontologías categóricas.** Estudian las diversas formas en las que una categoría da cuenta de los diversos niveles ontológicos, determinando la posible presencia de una teoría general que subsuma sus concretizaciones.

- **Ontologías del dominio.** Se refieren a la estructuración detallada de un contexto de análisis con respecto a los subdominios que lo componen.
- **Ontologías genéricas.** Aparecen ligadas a corpus lingüísticos y léxicos conceptuales.
- **Ontologías regionales.** Analizan las categorías y sus conexiones de interdependencia para cada nivel ontológico (estrato o capa).
- **Ontologías aplicadas.** Estas ontologías son la aplicación concreta del entorno ontológico a un objeto específico.

El último criterio que queda por exponer es el del grado de formalidad. En este caso sólo existe la clasificación de Poli (2002):

- **Ontologías descriptivas**. Relacionadas con la recolección de información sobre los ítems del dominio analizado.
- **Ontologías formales**. Destilan, filtran, codifican y organizan los resultados de una ontología descriptiva.

LENGUAJES DE MARCADO

Los lenguajes de marcado (en inglés *markup languages*), también denominados lenguajes de etiquetas o marcas, se pueden definir como formas de codificar documentos, ya sea texto, imágenes o cualquier tipo de exposición, donde la información presente del documento es acompañada de marcas o etiquetas, que aportan información extra, como distribución de imágenes y texto, semántica explícita del texto, etc.

Para el ámbito de aplicación del presente proyecto, las marcas o etiquetas que se añadirán tendrán una finalidad semántica. Ellas serán las que proporcionen el significado (semántica) explícito necesario para que las

máquinas comprendan los documentos. Añadir estas etiquetas, en este caso llamadas anotaciones, es lo que se conoce como anotar un documento.

En las secciones siguientes se presentarán los lenguajes de marcado utilizados comúnmente: XML, RDF(S) y OWL.

XML

XML proviene de *Extensible Markup Language*, o en español "lenguaje de marcas extensible". Según el W3C (http://www.w3.org/) es un sencillo formato de texto muy flexible derivado de SGML (ISO 8879). Originalmente diseñado para afrontar los retos a gran escala de la publicación electrónica, XML también está desempeñando un papel cada vez más importante en el intercambio de una amplia variedad de datos en la web y en otros entornos (http://www.w3.org/XML/). XML también se puede definir como un metalenguaje con el que se puede generar otros lenguajes, cuya gramática puede definirse gracias a sus etiquetas.

XML intenta estructurar la información de forma abstracta. Al estructurar los documentos, se obtienen secciones (elementos) muy bien diferenciadas, que a su vez se pueden escindir en más secciones. Estas secciones se señalizan mediante etiquetas. Todos los elementos tienen una etiqueta de inicio y una etiqueta de finalización o una etiqueta de elemento vacío, y deben estar dentro del mismo elemento.

Un objeto de datos es un documento XML bien formado. Un documento XML está bien formado si: tomado como un todo, concuerda con la producción de la gramática de XML marcada como documento; cumple con todas las restricciones acerca de buena formación contenidas en la especificación http://www.w3.org/TR/2008/REC-xml-20081126/; y, por último, cada una de las entidades procesadas, referenciadas directa o indirectamente en el documento, está, asimismo, bien formada. Concordar

con la producción documento de XML implica: (1) que contenga uno o más elementos (siempre existe un elemento llamado raíz o elemento documento, el cual no tiene parte alguna contenida en ningún otro elemento) y (2) que los elementos, delimitados por etiquetas de comienzo y de fin, se aniden apropiadamente (las etiquetas de inicio y fin deben estar en el contenido del mismo elemento). Además de estar bien formado, un documento XML es válido si tiene una "declaración de tipo de documento" o Document Type Declaration (DTD) asociada y el documento cumple con las restricciones indicadas en ella. Esta DTD, en caso de existir, debe aparecer, en el documento antes del primer elemento.

Desde el punto de vista de sus estructuras lógicas, un documento XML contiene uno o más elementos. Los límites de estos elementos son delimitados por etiquetas de inicio y fin, o, para elementos vacíos, por una etiqueta de elemento vacío. Cada uno de los elementos tiene un tipo, identificado por un nombre, a veces llamado "identificador genérico" (GI), y puede tener un conjunto de especificaciones de atributos. Cada especificación de atributo tiene un nombre y un valor.

Desde el punto de vista de sus estructuras físicas, un documento XML puede contar con una o más unidades de almacenamiento. Estas unidades se llaman entidades. Todas las unidades pose un contenido (excepto la entidad documento y el subconjunto de DTD) y todas son identificadas por el nombre de entidad. Cada documento XML tiene una única entidad llamada entidad documento, la cual sirve como punto de inicio para el procesador XML y puede contener el documento entero. Las entidades pueden ser *parsed* o *unparsed*. Los contenidos de una entidad *parsed* son su texto de remplazo. Este texto es considerado una parte integral del documento. Una entidad *unparsed* es un recurso cuyos contenidos pueden ser o no texto, y, si lo son, pueden no ser expresados en XML. Cada entidad *unparsed* tiene una notación asociada, identificada por un nombre. Aparte de requerir que un procesador

de XML debe hacer disponibles los identificadores de la entidad y la notación a la aplicación, XML no impone más restricciones para los contenidos de entidades *unparsed*.

RDF(S)

RDF (Resource Description Framework) es un lenguaje de representación de la información relativa a los recursos disponibles en la World Wide Web. Está especialmente pensado para la representación de los metadatos de recursos web, tales como su título, su autor, y la fecha de modificación de una página web y el copyright de un documento web, o la disponibilidad de algunos recursos compartidos. Sin embargo, al generalizar el concepto de "recurso web", RDF puede también ser usado para representar información sobre cualquier elemento que puede ser identificado en la web, incluso cuando este no pueda ser directamente recuperado en la web. Esto incluiría, por ejemplo, la información sobre los elementos disponibles desde una tienda on-line (*e.g.*, información sobre especificaciones, precios, y disponibilidad), o la descripción de las preferencias de un usuario web para una entrega.

RDF proporciona interoperabilidad entre aplicaciones que intercambian información procesable por máquinas en la web. RDF destaca por la facilitar el procesamiento automatizado de los recursos de la web. RDF puede aplicado en distintas áreas; por ejemplo: la en *recuperación de recursos,* para proporcionar mejores prestaciones a los motores de búsqueda; en *catalogación,* para describir el contenido y las relaciones de contenido disponibles en un sitio Web, una página Web, o una biblioteca digital particular; y por los *agentes de software inteligentes,* para facilitar el intercambio y para compartir conocimiento. También es de utilidad en la *calificación de contenido*, en la descripción de *colecciones de páginas* que representan un "documento" lógico individual, para describir los *derechos de*

propiedad intelectual de las páginas web, y para expresar las *preferencias de privacidad* de un usuario, así como las *políticas de privacidad* de un sitio Web.

El objetivo general de RDF es definir un mecanismo para describir recursos que no implique ninguna presunción sobre un dominio de aplicación particular, ni defina (a priori) la semántica de ningún dominio de aplicación. La definición del mecanismo debe ser neutral con respecto al dominio; sin embargo, el mecanismo debe ser adecuado para describir información sobre cualquier dominio.

Para facilitar la definición de metadatos, RDF cuenta con un sistema de clasificación muy parecido a los sistemas de programación y modelado orientado a objetos. Una colección de clases, producida normalmente para un propósito o dominio específico, se denomina esquema (*schema*). Las clases se organizan en una jerarquía, y son extensibles mediante el refinamiento de sus subclases. Así, para crear un esquema ligeramente diferente de uno existente no es necesario "reinventar la rueda", pues se pueden añadir modificaciones incrementales al esquema base. A través de la compartición de esquemas, RDF soporta la reutilización de definiciones de metadatos. Debido a la extensibilidad incremental de RDF, los agentes que procesen metadatos serán capaces de reconstruir los esquemas con los que no estén familiarizados, procesarlos, y realizar acciones significativas en los metadatos para cuyo proceso no fueron diseñados inicialmente. La capacidad de compartición y la extensibilidad de RDF permiten a los creadores de metadatos usar múltiples cambios de las categorías de objetos para "mezclar" definiciones, y para proporcionar múltiples presentaciones de sus datos, haciendo uso del trabajo de otros. En síntesis, es posible crear objetos específicos de datos basados en diversos esquemas de distintas fuentes (*i.e.*, "intercalando" diferentes tipos de metadatos). Los esquemas pueden estar escritos asimismo en RDF; existe un documento que describe cómo usar RDF para describir un vocabulario RDF. Este documento se puede encontrar en

http://www.w3.org/TR/rdf-schema/, y detalla la especificación de RDFS, descrita más adelante.

El fundamento o base de RDF es un modelo para representar propiedades de los recursos y los valores de estas propiedades. El modelo RDF se asemeja bastante al tradicional modelo de pares atributo-valor.

El modelo de datos básico de RDF consiste en los tres tipos de objetos que se describen a continuación:

- **Recursos:** todos los elementos descritos por expresiones RDF se denominan *recursos*. Un recurso puede ser una página web completa, tal como el documento HTML http://www.w3.org/Overview.html, por ejemplo; puede ser asimismo una parte de una página web; *e.g.* un elemento HTML o XML específico dentro del documento fuente; puede ser también una colección completa de páginas, *e.g.* un sitio web completo; o un objeto que no sea directamente accesible online, *e.g.* un libro impreso. Los recursos se designan siempre por sus URIs encerrados (opcionalmente) entre '<' y '>'. Cualquier cosa puede tener un URI; la extensibilidad de URIs permite la introducción de identificadores para cualquier entidad imaginable.
- **Propiedades:** Una *propiedad* es un aspecto específico, una característica, un atributo, o una relación que se usa para describir un recurso. Para cada propiedad se define su significado específico, sus valores permitidos, los tipos de recursos que puede describir, y sus relaciones con otras propiedades.
- **Sentencias:** Un recurso específico, junto con una propiedad definida, más el valor de dicha propiedad para ese recurso, es una *sentencia RDF* (RDF statement). Estas tres partes

individuales de una sentencia son, respectivamente, el sujeto, el predicado y el objeto de la sentencia. El objeto de una sentencia (es decir, el valor de la propiedad) puede ser otro recurso (especificado por un URI) o pude ser un literal (es decir, una cadena simple de caracteres u otro tipo de datos primitivo definidos por XML). En términos de RDF, un *literal* puede incluir como su contenido etiquetas XML pero estas ya no pueden ser procesadas más por un *parsser* de RDF. Existen varias restricciones sintácticas para expresar el etiquetado de literales en RDF.

Por su parte, RDFS (RDF Schema) es una especificación que describe cómo usar RDF para describir vocabularios RDF. Además, define también un vocabulario básico para el mismo. El documento del W3C (http://www.w3.org/TR/rdf-schema/) especifica el mecanismo de RDFS como un conjunto de recursos RDF (incluyendo sus clases y sus propiedades), y las restricciones que caracterizan sus relaciones. A continuación se presenta un pequeño listado con algunos de los vocabularios más importantes para clases y propiedades.

Como vocabulario concerniente a las clases se contempla el mostrado a continuación:

- `rdfs:Resource`-> Todo aquello que se describa con una expresión RDF se denomina recurso, y se considera una instancia de la clase `rdfs:Resource`.
- `rdf:Property`-> Representa el subconjunto de recursos RDF que son propiedades.
- `rdfs:Class`-> Corresponde al concepto genérico de un tipo o categoría, semejante a la noción de Clase en los lenguajes de programación orientados a objetos, tales como Java. Cuando un esquema define una nueva clase, el recurso que representa esa

clase debe tener una propiedad `rdf:type`, cuyo valor es el recurso `rdfs:Class`.

Como vocabulario concerniente a las propiedades se contempla el mostrado a continuación:

- `rdf:type` -> Indica que un recurso es miembro de una clase, de tal forma que tiene todas las características propias de los miembros de esa clase. Cuando un recurso tiene una propiedad `rdf:type` cuyo valor es alguna clase específica, decimos que el recurso es una instancia de la clase especificada. El valor de una propiedad `rdf:type`, para algunos recursos es otro recurso, que tiene que ser instancia de `rdfs:Class`.
- `rdfs:subClassOf`-> Esta propiedad especifica una relación subconjunto/superconjunto entre clases. La propiedad `rdfs:subClassOf` es transitiva. Si la clase A es una subclase de otra clase B más amplia, y B es una subclase de C, entonces A es también implícitamente una subclase de C. Por lo tanto, los recursos que son instancias de la clase A serán también instancias de C, puesto que A es un subconjunto de ambas, tanto de B como de C. Sólo los objetos específicos de una instancia de `rdfs:Class` pueden tener la propiedad `rdfs:subClassOf`, y el valor de la propiedad es siempre `rdf:type rdfs:Class`. Una clase puede ser subclase de más de una clase.
- `rdfs:subPropertyOf`-> La propiedad `rdfs:subPropertyOf` es una instancia de `rdf:Property` que se utiliza para especificar que una propiedad es una especialización de otra. Una propiedad puede ser una especialización de cero, una, o más propiedades. Si alguna propiedad, P2, es una `subPropertyOf` (especialización de) otra propiedad más general, P1, y si un

recurso, A, tiene la propiedad P2 con un valor B, esto implica que el recurso A tiene también una propiedad P1 con valor B.

- `rdfs:range`-> `rdfs:range` es una instancia de `rdf:Property` usada para especificar que los valores de una propiedad son instancias de una o más clases. La tripleta `P rdfs:range C` establece que P es una instancia de la clase `rdf:Property`, que C es una instancia de la clase `rdfs:Class`, y que los recursos denotados por los objetos de las tripletas cuyo predicado es P son instancias de la clase C.
- `rdfs:domain`-> `rdfs:domain` es una instancia de `rdf:Property` que es usada para especificar que cualquier recurso referido por una propiedad es una instancia de una o más clases. La tripleta `P rdfs:domain C` establece que P es una instancia de la clase `rdf:Property`, que C es una instancia de la clase `rdfs:Class` y que los recursos denotados por los sujetos de las tripletas cuyo predicado es P son instancias de la clase C.

OWL

La información expuesta en este apartado, que proporciona una pequeña introducción a OWL, ha sido obtenida de la recomendación del W3C disponible en http://www.w3.org/TR/owl-features/ y de su traducción oficial al español (http://www.w3.org/2007/09/OWL-Overview-es.html).

OWL es el acrónimo de las palabras inglesas Ontology Web Language, "Lenguaje Web de Ontologías". OWL está pensado para ser usado cuando la información contenida en los documentos necesita ser procesada por las aplicaciones, al contrario que en las situaciones en las que el contenido sólo necesita ser presentado a los humanos. OWL puede ser usado para representar explícitamente el significado de los términos de un vocabulario y

las relaciones entre los mismos. Esta representación de términos y sus interrelaciones constituye una ontología. OWL se basa en RDF(S), pero tiene mayor capacidad para expresar el significado y la semántica de los terminos que XML, RDF, y RDF-S. De este modo, OWL va más allá que estos lenguajes en su capacidad para representar el contenido de la web interpretable por un ordenador. Además, como se construye a partir de estos lenguajes, tiene la misma potencia de expresión que todos ellos, más las nuevas características añadidas. OWL es una revisión, realizada por el W3C, del Lenguaje de Ontologías Web DAML+OIL que incorpora las lecciones aprendidas a partir de su diseño y aplicación.

OWL proporciona tres sublenguajes, cada uno con un nivel de expresividad mayor que el anterior, diseñados para ser usados por comunidades específicas de desarrolladores y usuarios: OWL Lite, OWL DL y OWL Full. ***OWL Lite*** está diseñado para aquellos usuarios que necesitan principalmente una clasificación jerárquica y restricciones simples. Por ejemplo, aunque admite restricciones de cardinalidad, sólo permite establecer valores para la misma de 0 ó 1. ***OWL DL*** está diseñado para aquellos usuarios que, conservando completitud computacional (se garantiza que todas las conclusiones sean computables), y decidibilidad (todos los cálculos se resolverán en un tiempo finito), quieran la máxima expresividad posible. OWL DL incluye todas las construcciones del lenguaje de OWL, pero sujetas a ciertas restricciones (por ejemplo, mientras una clase puede ser una subclase de otras muchas clases, una clase no puede ser una instancia de otra). OWL DL es denominado de esta forma debido a su correspondencia con la Lógica Descriptiva (**D**escription **L**ogics, en inglés), un campo de investigación que estudia la lógica que compone la base formal de OWL. ***OWL Full*** está dirigido a usuarios que buscan la máxima expresividad y libertad sintáctica de RDF sin importarles tanto los aspectos computacionales.

A continuación se introducen algunas de las características nuevas que provee OWL Lite con respecto a RDF(S). Aquellas de las que dispone por estar basado en RDF(S), como rdfs:domain se han expuesto ya anteriormente.

- Class: Una clase define un conjunto de individuos que pertenecen al mismo porque comparten algunas propiedades. Las clases pueden organizarse en una jerarquía de especialización usando subClassOf. Se puede encontrar una clase predefinida llamada Thing, que es la clase de todos los individuos y es una superclase de todas las clases de OWL. También se puede encontrar otra clase predefinida llamada Nothing, que es la clase que no tiene instancias y es una subclase de todas las clases de OWL.
- Individual: Los individuos son instancias de clases. Los individuos pueden relacionarse entre sí mediante propiedades. Por ejemplo, un individuo llamado *Deborah* puede ser descrito como una instancia de la clase *Persona*, y la propiedad *esEmpleadoPor* puede ser usada para relacionar el individuo *Deborah* con el individuo *UniversidadDeStanford*.
- equivalentClass: Define dos clases como equivalentes. Las clases equivalentes tienen las mismas instancias. El valor de igualdad puede ser utilizado para crear clases sinónimas.
- equivalentProperty: Define dos propiedades como equivalentes. Las propiedades equivalentes relacionan a un individuo con el conjunto de otros individuos similares. Por ejemplo, *tieneLíder* podría definirse como equivalentProperty (propiedad equivalente) de *tienePresidente*.
- sameAs: Define a dos individuos como iguales. Esta construcción pueden utilizarse para crear un número de nombres diferentes que se refieren al mismo individuo.

- differentFrom: Define a un individuo como diferente de otro(s) individuo(s). Por ejemplo, el individuo *Frank* puede establecerse como distinto de los individuos *Deborah* y *Jim*.
- AllDifferent: Indica que todos los individuos son diferentes entre sí. Por ejemplo, se podría establecer que Frank, Deborah y Jim son mutuamente distintos usando la construcción AllDifferent. Al contrario que la indicación *differentFrom* anterior, esto además resaltaría que *Jim* y *Deborah* son distintos (no únicamente que *Frank* es distinto de *Deborah* y *Frank* es distinto de *Jim*).
- minCardinality: Establece la cardinalidad mínima de una propiedad para una clase específica. Por ejemplo, la propiedad *tieneDescendiente* no requiere una cardinalidad mínima para la clase *Persona*, y que una persona no necesariamente tiene hijos. Pero esta misma propiedad sí debería tener cardinalidad mínima uno para la clase *Progenitor*. Como restricción para OWL Lite, la propiedad minCardinality sólo puede tener cardinalidad cero o uno.
- maxCardinality: Establece la cardinalidad máxima de una propiedad para una clase específica.
- cardinality: Esta propiedad sirve para indicar que una propiedad tiene el mismo valor para la cardinalidad mínima (minCardinality) como para la máxima (maxCardinality).
- intersectionOf: Esta propiedad sirve para definir una clase como la intersección de otras dos clases ya definidas. Por ejemplo, la clase *PersonaEmpleada* se puede describir como la *intersectionOf* (intersección entre) *Persona* y *ObjetosEmpleados* (que podrían ser definidos como objetos que tienen una cardinalidad mínima de 1 en la propiedad *tieneJefe*). A partir de

esto, un razonador podría deducir que cualquier *PersonaEmpleada* tiene por lo menos un jefe.

JENA

Jena es un entorno de desarrollo en Java cuyo objetivo es la construcción de aplicaciones para la Web Semántica. En la dirección web http://jena.sourceforge.net/ está disponible el código, la licencia y la documentación de la misma. Proporciona un entorno de programación para trabajar con los lenguajes RDF, RDFS, OWL y SPARQL. Además, incluye un motor de inferencias basado en reglas. De esta forma se puede trabajar con RDF, RDFS y OWL, obteniendo sus modelos, representados por objetos; realizar consultas sobre estos modelos gracias al lenguaje de consulta SPARQL; y razonar sobre las sentencias de los modelos, obtener conclusiones y ampliar el conocimiento disponible.

Con Jena se pueden leer los archivos que contienen ontologías, construyendo los correspondientes modelos, que serán encapsulados en objetos Java. Una vez que se tienen en memoria las ontologías, se puede acceder a sus sentencias, recursos, predicados, literales, namespaces,etc., y a todos los elementos de los que se componen los documentos redactados en RDF y OWL. Tras la lectura, se puede navegar por el modelo buscando las instancias, recursos o predicados deseados, o comprobar si existen. Se pueden modificar estos modelos como sea necesario, añadiendo, eliminando o cambiando las sentencias. Finalmente, una vez realizadas las modificaciones, se puede escribir el modelo en un archivo del disco.

ANÁLISIS DETALLADO DE LAS HERRAMIENTAS EXISTENTES PARA LA ANOTACIÓN SEMÁNTICA

Las herramientas de anotación basadas en ontologías, alias de los anotadores basados en ontologías, son ante todo diseñadas para permitir la inserción de etiquetas provenientes de ontologías, en páginas web. La mayoría de estas herramientas han aparecido recientemente con el nacimiento de la Web Semántica. Los anotadores fueron concebidos inicialmente para aliviar la carga de incluir manualmente anotaciones basadas en ontologías en páginas web. Desde entonces, muchas de ellas han evolucionado en entornos más completos que usan técnicas de Extracción de Información, en inglés Information Extraction (IE), y Aprendizaje Automático, en inglés Machine Learning (ML), posponiendo y realizando anotaciones semiautomáticas para documentos web (Corcho, 2006).

En este apartado, se analizarán algunas de las herramientas que, en un principio, se pensó que podrían cubrir las necesidades por las que nace el proyecto. Tras una primera criba, donde se eliminaron ya varias herramientas, quedaron cuatro: OntoMat-Annotizer, AKTive Media, MnM y ONTO-H. Después de ahondar en las características de las cuatro herramientas, se mostrará que la que más se ajustaba a dichas necesidades era AKTive Media, pero que requería ser ampliada con nuevas funciones y características, que se detallarán en el apartado de Desarrollo.

OntoMat Annotizer (http://annotation.semanticweb.org/ontomat/index.html) es una herramienta del proyecto OntoAgents (http://infolab.stanford.edu/OntoAgents/) sencilla y manejable para anotar páginas web o documentos en HTML. Su sencillez radica en su modo de uso, *drag and drop* (traducido al español, arrastrar y soltar). Con este modo de uso, prácticamente cualquier tipo de usuario sin conocimientos previos puede llevar a cabo su tarea sin mucho esfuerzo.

Los documentos que toma como entrada, y sobre los que realiza las anotaciones tienen que estar escritos en lenguaje HTML. Estos pueden ser tanto ficheros locales como páginas de la WWW, especificadas mediante su URL.

Las ontologías con las que trabaja la herramienta tienen que estar descritas en el lenguaje DAML+OIL. Seleccionando un fragmento de texto y asociándolo a un concepto de los existentes en la ontología, se creará una instancia de dicho concepto.

En cuanto al formato y el modo en el que se almacenan las anotaciones realizadas, estas son empotradas en el código HTML de los archivos de entrada a modo de comentarios. Tanto si el documento de entrada es un fichero local, como si es una página de la WWW, la herramienta crea un nuevo fichero local, escrito en HTML, con las anotaciones realizadas. En el caso de los ficheros locales, el fichero de salida será una copia del fichero de entrada con las anotaciones como comentarios. En el caso de las páginas web, el fichero de salida será el resultado de transformar, la página en código HTML más las anotaciones realizadas, añadidas también como comentarios.

Las anotaciones, una vez realizadas, no se visualizan en el documento, ya que solamente queda constancia de la anotación en el fichero de salida, circunstancia que dificulta su reconocimiento y recuperación, con el

consiguiente problema de no poder diferenciar dos anotaciones para el mismo texto.

La herramienta es independiente de la plataforma y del sistema operativo, y está escrita en Java, un aspecto muy importante, ya que se desea construir una aplicación multiplataforma. Además, la herramienta se distribuye bajo licencia GNU Lesser General Public License (LGPL). Por lo tanto, se podría optar por partir de esta herramienta y extenderla con todas las nuevas funcionalidades que se desean, como por ejemplo, distintas pestañas para los diferentes niveles de anotación, extender el formato de los archivos de entrada, etc.

Sin embargo, esta herramienta presenta un gran inconveniente: a pesar de ser código libre, no dispone de los ficheros fuentes para poder modificarlos y recompilarlos. Existen tres formas de distribución de la misma: a través de un único archivo, con extensión "jnlp", mediante archivos comprimidos con extensión "jar", que son llamados mediante un archivo de proceso por lotes (con extensión "bat") o bien con los archivos fuente compilados.

En el caso del archivo con extensión "jnlp", que ejecuta el programa en un servidor web. Basta con hacer doble clic en el icono y éste se ejecuta de forma remota. Esta forma de utilizar la aplicación puede ser muy útil, ya que no se requiere de instalación previa, evitando problemas. Por contra, imposibilita el acceso a los ficheros fuente y, consecuentemente, su modificación.

Las alternativas de distribución de los archivos con extensión "jar" y "class" tampoco permiten reutilizar el código de la aplicación. A pesar de ello, se barajó la posibilidad de seguir un proceso de ingeniería inversa, para que, a partir de los archivos con extensión "class", se pudiesen obtener los ficheros con extensión "java" y así poder utilizar partes de la aplicación. Para realizar la ingeniería inversa y conseguir los archivos fuente, se utilizarón tanto el comando *javap*, el desensamblador propio de Java, como alguno de los

muchos programas existentes con esa finalidad. Mediante el comando *javap* no se consiguió código fuente en Java, sino un pseudocódigo de escasa o nula utilidad. Básicamente, la información resultante se reducía a los nombres y los tipos de las variables que se utilizan en el programa, así como los nombres de sus métodos con sus respectivos parámetros, tanto de entrada como de salida. Con otras herramientas, como Cavaj Java Decompiler (http://www.sureshotsoftware.com/cavaj/index.html), el proceso hubo de hacerse archivo a archivo y, si bien se recuperó parte del código, otra parte, la más complicada en realidad (porque hace referencia a otras clases o hace uso de clases anónimas), no se pudo traducir.

AKTIVE MEDIA

AKTive Media (http://www.dcs.shef.ac.uk/~ajay/html/cresearch.html) es un sistema basado en ontologías, para la anotación tanto de imágenes como de texto. Su objetivo principal es ayudar al usuario en el proceso de anotación, interactuando con el mismo y haciéndole sugerencias, mientras el usuario va anotando el documento, por lo que se reduce su esfuerzo. Los distintos formatos admitidos para las ontologías con las que se anota usando la herramienta son: OWL, DAML y RDFS.

Existe la posibilidad de anotar tanto texto como imágenes, y en el caso de que se trate de documentos HTML, una combinación de ellas. Los tipos de archivos de entrada que son admitidos por la herramienta son texto plano, HTML e imágenes (en los formatos JPG, GIF, BMP, PNG y TIFF). Las anotaciones de cada documento se almacenan en un archivo RDF con el mismo nombre que el archivo origen, y dentro de una estructura de carpetas determinada, dentro de la sesión abierta. Las operaciones de marcado del texto quedan definidas en el documento a través del resaltado del texto, de acuerdo con el color que tenga asociado el concepto con el que se ha realizado la operación. Con este método se obtiene, a primera vista, una lectura rápida y cómoda del documento anotado. Esto contribuye también a que el interfaz resulte mucho más amigable y llamativo.

Para la anotación de imágenes, existe también la posibilidad de realizar un anotado por lotes (en inglés, *batch annotation*), con lo que se consigue anotar una colección completa de imágenes a la vez.

La estructuración de los archivos de salida se compone de sesiones, en la cuales puede coexistir más de un documento para su anotación, pero solamente admite la importación de una única ontología para ello.

MnM (http://kmi.open.ac.uk/projects/akt/MnM/index.html) (Corcho, 2006) es una herramienta autónoma que integra un navegador web y un editor de ontologías, y que proporciona unas APIs abiertas para interconectar MnM con servidores de ontologías y con herramientas de extracción de información. Permite anotar documentos de tres formas distintas: manual, semiautomática y automática. Ha sido desarrollado por el Knowledge Media Institute en la Open University (UK), en el contexto de AKT Interdisciplinary Research Collaboration (http://www.aktors.org/).

MnM es una aplicación Java extensible, basada en una arquitectura *plugin*, disponible para la descarga desde la mencionada URL. Admite la carga de ontologías gestionadas por un servidor de ontologías, almacenadas en archivos o mediante una URL, escritas en cualquiera de los siguientes lenguajes para describir ontologías: RDF(S), OWL, y OCML. De forma similar, las anotaciones creadas con esta herramienta pueden ser usadas para poblar ontologías existentes o ser adjuntadas al documento original (en formato XML, donde los nombres de las etiquetas son los nombres de los conceptos, de sus atributos, y de sus relaciones).

En cuanto a la anotación automática de documentos, MnM usa motores de extracción de información para detectar la existencia de instancias de conceptos en los documentos. Estos motores deben ser entrenados con un conjunto de textos y documentos HTML anotados, a fin de que generen las reglas utilizadas para extraer la información de otros documentos. Cuando el modulo está entrenado, este puede ser usado para detectar instancias de conceptos, valores de atributos e instancias de relaciones en otros documentos. Los usuarios pueden decidir modificar las anotaciones realizadas por el módulo que extrae la información o dejarlas como han sido generadas.

En la distribución estándar de la herramienta, se incluye un *plugin* con el motor de extracción de información Amilcare (Ciravegna, 2001) es incluido. También podrían añadirse otros motores para la extracción de información como *plugins*.

Las anotaciones generadas por esta herramienta pueden ser usadas en diferentes entornos. MnM almacena instancias en varios formatos: OCML (de esta forma pueden ser usadas por cualquier herramienta o aplicación de conocimiento OCML como WebOnto, Planet-Onto, etc.), RDF, OWL, y XML.

Onto-H (Benjamins et al., 2004) es un plugin del editor de ontologías Protégé que permite crear anotaciones de documentos RTF (Corcho, 2006). Este plugin ha sido desarrollado por iSOCO (http://www.isoco.com/) en el contexto del proyecto europeo Esperonto.

Ya que ONTO-H está integrado en el editor Protégé, este reutiliza muchas de sus funciones, tales como el buscador de ontologías, el cual es similar a la pestaña *Classes&Instance* que está presente en la distribución por defecto de Protégé. Además, los usuarios de ONTO-H pueden reutilizar todas las funciones proporcionadas por el editor de Protégé, tales como el editor de ontologías, el navegador de funciones, visualizador de ontologías, el mezclador, etc. Entre ellas cabe destacar, todas las funciones de importación y exportación del editor, que brindan una gran flexibilidad con respecto a los formatos en los que las anotaciones serán almacenadas.

La interfaz de usuario de la herramienta hace uso de *drag & drop* (en español "arrastrar y soltar"). Seleccionando conceptos de una ontología y soltándolos en el panel de instancias, se crean nuevas instancias de esos conceptos. Además de las funciones *drag & drop* para crear anotaciones manualmente, el editor también proporciona sugerencias para la anotación de partes del texto, mediante el reconocimiento de entidades con nombre, anotaciones ya existentes con el mismo nombre o con un sinónimo, etc. En este sentido, ONTO-H es una herramienta que puede ser principalmente usada para anotación supervisada, en vez de para un proceso de anotación completamente manual.

Por último, ONTO-H permite el uso de reglas declarativas implementadas en el lenguaje de reglas DROOLS (http://www.drools.org/). Estas reglas son usadas para indicar al usuario automáticamente, con ejemplos, formas de edición que permiten la creación de nuevas instancias

relacionadas con la que se acaba de soltar sobre el panel de instancias. Esta función ha demostrado ser muy útil en el dominio de la cultura, donde instancias de una obra de arte realizada por un artista están, la mayoría de las veces, relacionadas con instancias de expresiones y manifestaciones de tal obra.

CONCLUSIONES SOBRE LAS HERRAMIENTAS

En este apartado se argumentará por qué se decide proseguir el desarrollo con la herramienta AKTive Media y por qué se descartaron las otras tres.

La herramienta OntoMat Annotizer fue la primera en ser descartada. Esta herramienta fue la que más aspectos negativos acumulaba. Aunque estos aspectos negativos no eran muy significativos, sí la penalizaban frente a las otras. En cualquier caso, hasta que no se intentó estar en posesión de su código no se encontró el problema más importante y que la descartó al poco tiempo: la disponibilidad del código. A pesar de que se intentó obtener su código, como se comentó al analizarla, no se consiguió. Por lo tanto se desechó como alternativa. Una lista resumen de sus aspectos negativos es la siguiente:

- Sólo admitía un único formato de entrada para las ontologías: DAM+OIL.
- Las anotaciones eran empotradas en una copia del documento inicial.
- Las anotaciones no se visualizaban en el texto mientras se anotaba.
- No era posible obtener su código fuente para su reutilización y extensión con nuevas funciones.

También hubo una razón importante para desechar ONTO-H. Esta herramienta, en realidad, era un *plugin* incluido en la distribución estándar de la herramienta Protégé (http://protege.stanford.edu/), un editor de ontologías, y no se deseaba tener que ceñirse a un editor de ontologías determinado.

Una vez se descartaron esta dos opciones, sólo quedaban MnM y AKTiveMedia. Estas dos herramientas eran muy parecidas en cuanto a funcionalidad y se optó por AKTiveMedia por ser de más fácil manejo, por requerir de un proceso de configuración más sencillo y por tener una interfaz más atractiva.

OBJETIVOS DEL PROYECTO

El propósito del presente proyecto era desarrollar una herramienta de anotación semántica. Esta herramienta debía tomar como etiquetas para sus antoaciones las ontologías. Además, debía amoldarse a un modelo predefinido de ontologías. Este modelo de ontologías tiene su origen en el campo de la Lingüística de Corpus (LC). Por lo tanto al basarse tanto en ontologías como en la LC, la herramienta resultante será un híbrido entre la anotación semántica de la Inteligencia Artificial y de la Lingüística de Corpus. Este modelo híbrido de anotación se basa principalmente en el uso de cuatro ontologías lingüísticas: la ontología de unidades, la de atributos, la de valores y la de relaciones. El uso de cada una de estas ontologías se describe a continuación. Cada anotación realizada sobre el texto debía pertenecer a un concepto de la ontología de unidades. De esta forma, dependiendo del tipo de concepto al que pertenezca la anotación, esta tendrá unos u otros atributos, que están disponibles en la ontología de atributos. Como es lógico, estos atributos tomarán unos valores, que están determinados en la ontología de valores. Y, para concluir la descripción de las ontologías, las unidades se pueden relacionar unas con otras, estando delimitadas en la ontología de relaciones todas las posibles asociaciones entre las unidades.

Debido a su posible uso por lingüistas, la herramienta debía tener una interfaz sencilla, ya que estos usuarios pueden no estar familiarizados con temas informáticos. Por consiguiente la interfaz debía incorporar la posibilidad (entre otras) de anotar los documentos mediante la técnica *drag and drop*, en español “arrastrar y soltar”. Esta técnica se fundamenta en seleccionar componentes visuales, arrastrarlos y soltarlos sobre algún otro componente, para que se produzca el efecto deseado.

DESARROLLO DEL PROYECTO

En este apartado se detallarán los aspectos relativos al desarrollo del proyecto, es decir, el proceso por el cual, a partir de la aplicación inicial, AKTive Media (versión 1.9), se consigue llegar a la nueva aplicación final, OntoLing Annotizer, mediante modificaciones de funciones y añadidos. Gracias a las modificaciones y añadidos, se consigue alcanzar los objetivos y cubrir las expectativas de la nueva herramienta. La estructura que se seguirá en cada caso consistirá en la descripción del problema que tiene la herramienta de partida y hay que solucionar, y a continuación, se mostrará la solución adoptada. Dicha solución se detallará de forma conceptual y en líneas generales, siempre que la explicación no sea de muy bajo nivel y, por lo tanto, no sea relevante para la documentación del proyecto.

1) El primer problema se produjo al compilar los archivos fuente escritos en lenguaje Java. Al compilar el proyecto en su totalidad se producía un error, provocado por la inexistencia de una clase que forma parte del mismo. La versión que se intentaba compilar fue la versión 1.9.

   La primera solución considerada consistió en mirar el código fuente de versiones anteriores, con la esperanza de encontrar la clase ausente en la última versión. A medida que se descendía en la lista de versiones, estas disponían de una menor funcionalidad y como consecuencia, estaban constituidas por menos clases. Entre las clases que se habían desarrollado para la versión actual se encontraba la clase deseada, por lo que no existía en las anteriores versiones. Así que se desechó esta posibilidad.

   Otra solución que se consideró fue realizar ingeniería inversa, utilizando una de las herramientas llamadas “decompiladores” (*decompiler* en inglés). Con estas herramientas, a partir del código

objeto, se puede llegar a obtener una aproximación del código fuente. La bondad de la aproximación a la que se llega depende en gran medida, de lo complicado que sea el código. Esta dificultad en el código se mide, principalmente, mediante el número de referencias a otras clases. De los compiladores accesibles se optó por CAVAJ JAVA DECOMPILER, por ser una utilidad freeware y por su fácil e intuitivo manejo.

El proyecto constituido por los archivos fuente no estaba operativo, pero el proyecto constituido por los archivos objeto ya compilados (en Java se identifican por la extensión ".class") sí lo estaba y contenía todas las clases de las que se requería para un correcto funcionamiento. Traduciendo los archivos objeto que producían el error al lenguaje fuente (Java), se descubrió en qué paquete debía estar contenida la clase. Traduciendo la clase que faltaba de código objeto a código fuente, se ubicó en ese paquete. De esta forma, se pudo resolver el problema y se pudo partir de un proyecto operativo, que se podía compilar y ejecutar.

2) Una vez que se dispuso de una herramienta plenamente operativa, se sometió a una evaluación previa para confirmar sus prestaciones y comprender claramente su funcionamiento. La primera vez que cargó una de las ontologías con las que se va a realizar el anotado, se constató que no se recuperaba la ontología en su totalidad, sino tan solo unas pocas clases.

   En efecto, el árbol que mostraba los conceptos de la ontología no estaba completo: le faltaban la mayoría de las clases. Por este motivo, se tuvo que cambiar el tratamiento que se hacía de la ontología en el momento de recuperarla y cargarla. Uno de los principales fallos era que no se tenían en cuenta las propiedades, y estas, en el modelo

ontológico, se representan como superclases de los conceptos, a través de un nodo anónimo.

3) Se observó que los colores disponibles para ser asignados a los conceptos de las ontologías eran muy escasos. Solo se disponía de diecinueve colores, y una vez que se alcanzaba este número de conceptos, se repetía la lista los colores.

   Para solucionarlo, se realizó un pequeño programa que generó una lista aleatoria de quinientos colores. Esta lista generada es la que se utiliza para asignar colores a los conceptos. Esta lista es la misma siempre: el carácter aleatorio sólo estuvo presente en el momento de generarla. Al igual que antes, si se alcanzan los quinientos un colores, se repite la secuencia. A la hora de generar la lista de colores, se intentó que no fuesen colores muy ni oscuros ni muy claros. Como los colores se crean mediante el modelo RGB, se evitaron los valores cercanos al cero y al doscientos cincuenta y cinco a la hora de generarlos.

4) Al solucionar el problema de la reconstrucción del árbol con el que el usuario interactúa con la ontología, se produjo un fallo colateral. Cada nodo del árbol, que representa un concepto de la ontología, tiene un color asignado, de forma que cuando se anota el texto con un concepto, queda resaltado el texto anotado con su color. Tras los cambios introducidos, cuando se anotaba con un concepto de los que antes no se recuperaban (que eran la mayoría) se producía un desajuste: no concordaba el color del concepto con el color con el que quedaba resaltada la unidad perteneciente al texto.

   Esto se solucionó cambiando la forma en que se asignaban los colores, ya que no se tenía en cuenta la repetición de conceptos en el árbol que representa la ontología. Existen conceptos repetidos

porque, en las ontologías de partida, un mismo concepto puede heredar de más de un concepto, es decir, presentan herencia múltiple. Pero el problema en sí no era este, sino que provenía de la estructura de representación. Como se tiene una estructura arbórea para visualizar cada una de las ontologías, la herencia múltiple conlleva la repetición de un concepto tantas veces como padres distintos tenga, y a cada repetición se le asignaba un color nuevo. Por lo tanto, asignando el mismo color a cada una de estas repeticiones se resolvió este problema.

5) El siguiente cambio necesario que se introdujo permitió poder utilizar más de una ontología por sesión y gestionarlas. Esto fue necesario porque la herramienta permitía anotar con una ontología únicamente por sesión, y la estructura de anotación ontológica que requería el proyecto necesitaba de varias ontologías.

   En lo que respecta al aspecto gráfico, en la pantalla inicial se pasó de tener un campo que sólo mostraba una ontología y un botón para cargarla, a tener una lista donde al cargarse se muestran todas las ontologías seleccionadas y dos botones: uno para añadir más ontologías y otro para quitarlas. Además, se sustituyó el panel que mostraba la ontología una vez iniciada la sesión, por un panel con pestañas, donde aparece una nueva pestaña por cada ontología que haya sido cargada. Para trabajar con cualquiera de las ontologías cargadas, basta con seleccionar la pestaña que contenga la ontología con la que se quiera trabajar. Las pestañas se pueden diferenciar porque poseen, como título el nombre, de la ontología que contienen.

   En cuanto a aspectos internos, ahora se tienen varias ontologías por sesión, cuando anteriormente se tenía solamente una, por lo que hay que almacenar más información por usuario. Hay que asegurarse

que, al iniciar una sesión, exista al menos una ontología asociada a dicha sesión.

6) Una consecuencia inmediata de la posibilidad de añadir más de una ontología por sesión, es la diferenciación entre conceptos con el mismo nombre, pero pertenecientes a ontologías distintas. Debido a la existencia de conceptos con el mismo nombre, no se podría saber a qué ontología pertenece una anotación en la que, como se venía haciendo hasta el momento, solo se almacene el nombre del concepto para poder identificarla. Esto último es necesario porque la información que proporcionan las distintas ontologías no es la misma. Por ejemplo, no sería lo mismo anotar una ocurrencia de banco con relación al concepto BANCO de una ontología del dominio mercantil que de una ontología del dominio del mobiliario urbano.

   La solución a este problema hace uso del espacio de nombres XML (en inglés, *XML namespace*) asociado a cada ontología, y que es necesariamente distinto para cada una. Los espacios de nombres XML proporcionan un método simple para cualificar los nombres de los elementos y los atributos usados en XML, asociándolos con espacios de nombres identificados por referencias a URIs. De esta forma, para cada anotación se almacena el nombre del concepto al que pertenece y el namespace de la ontología de la este que procede.

7) En este momento ya se podría cargar más de una ontología por sesión y, por lo tanto, se podían introducir los distintos tipos de ontologías. Como ya se mencionó, en total existen cuatro tipos de ontologías: de unidades, de atributos, de valores y de relaciones. Con la de unidades se crearán las anotaciones. Las anotaciones contienen fragmentos de los documentos HTML que se asocian a un concepto de este primer tipo de ontologías. Estas anotaciones incluyen también a dicho concepto y pueden poseer unos atributos, y estos

atributos se toman de las ontologías de atributos. A su vez, los atributos adquieren valores, que son tomados de las ontologías de valores, adjuntándose a la anotación correspondiente. Por último las unidades se pueden relacionar con otras unidades, mediante relaciones binarias, que son tomadas de las ontologías de relaciones.

La forma de implementar los distintos tipos de ontologías es tan sencilla como asignar un atributo a las ontologías donde se indique a que tipo pertenecen. De esta forma, en el momento de utilizar un concepto de una ontología basta con comprobar, primeramente, a qué tipo de ontología pertenece y, posteriormente, si es posible la operación que se va a realizar, de acuerdo a ese tipo de ontología.

8) En la función anterior, se han introducido los tipos de ontología que se manejan, y se ha comentado cuál es su finalidad. Ahora quedaba pendiente introducir en la herramienta la utilización de estos tipos de ontologías.

   Para las ontologías de unidades no es necesario ningún cambio, a excepción de sus restricciones de uso. Estas restricciones de uso indicaban que no se puede utilizar un tipo de ontología en lugar de otra. Por ejemplo, no se puede utilizar una unidad como valor de un atributo, lo cual es extensible a todos los tipos de ontologías. Para las ontologías de atributos y valores, se añadió un par a cada anotación por cada atributo que se asocie a una anotación con su correspondiente valor. De esta forma, cada anotación tendrá una lista de pares atributo-valor. Con las ontologías de relaciones se procedió de un modo parecido a como se hizo con las ontologías de atributos y valores, asociando un par por cada nueva relación. La diferencia de las ontologías de relaciones con las otras estriba en que uno de los elementos del par es también una unidad, con la posible duda de a cuál de las dos unidades asignarle el par. La solución

consistió en asignar la relación a la unidad origen de la relación y, para denotar la segunda anotación, anotación destino, se utiliza una referencia a la misma.

9) El noveno cambio consistió en la adicción de una nueva función en el momento de configurar una sesión. Se deseaba que existiera la posibilidad de cargar un conjunto de ontologías de una vez, activando un "check box". Este conjunto sería básico para empezar a trabajar, y estaría compuesto por cuatro ontologías, cada una de una clase diferente: unidades, atributos, valores y relaciones.

   Se barajaron dos posibles implementaciones. La misma consistía en destinar una carpeta específica para almacenar estas ontologías; la segunda se basaba en el uso de un fichero de configuración.

   La primera posibilidad, conllevaba varios inconvenientes. Por ejemplo, las ontologías contenidas en esta carpeta deberían tener unos nombres específicos para de esta forma, poder saber a qué tipo concreto (unidades, atributos, valores y relaciones) pertenece cada cual; si se quisiera cambiar este conjunto de ontologías, se tendrían que eliminar las que están en esta carpeta y mover las nuevas desde su ubicación actual a la carpeta mencionada anteriormente, requiriendo un mayor tráfico de información, además de resultar una molestia; daría lugar a redundancia e inconsistencia de la información, ya que muchas veces, el usuario desea conservar sus ontologías en su ubicación original; es poco intuitivo y visual para el usuario, lo que penaliza esta primera posibilidad.

   La segunda posibilidad consistía en crear un archivo de configuración donde se almacenara la dirección de estas ontologías. De esta forma se evitan los inconvenientes anteriores: de tráfico innecesario, pues no hay que transferir archivos; redundancia de

información, pues se indican las rutas de las ontologías y así no se tienen que duplicar. Además, existe la posibilidad de conservar los nombres que tuviesen los ficheros, o, si se quiere, modificar alguno o algunos, cambiarles el nombre. Un aspecto negativo podría ser que, al igual que la opción anterior, sigue sin ser quizás muy visual y amigable. Este problema se solucionó permitiendo modificar el archivo de configuración mediante un formulario, en el que hay una etiqueta, un campo y un botón por cada tipo de ontología. La etiqueta identifica qué campo y botón se asocian con cada tipo de ontología. El campo muestra el nombre de la ontología. Se muestra únicamente el nombre porque, si el usuario es un lingüista, que por lo general tendrá unos conocimientos básicos en informática, mostrar la ruta absoluta o relativa del archivo le puede confundir. Como se acaba de comentar, el usuario habitual de la herramienta puede desconocer el uso, incluso la existencia, de la ruta de los archivos. Por lo tanto, al pulsar un botón del formulario aparece un selector de archivos, con el que se trabaja de forma encubierta, para el usuario, con las rutas de los archivos.

10) Existe un requisito que proviene de la forma en la que vienen dadas algunas ontologías y, más específicamente, de cómo están descritas las instancias. Las instancias no están en el mismo archivo, en lenguaje OWL, que la ontología, sino que están repartidas en varios archivos distintos, pero dentro de la misma carpeta que contiene a la ontología, con extensión (y descritas en) RDF. Esto podría llegar a ser un problema, porque pertenecen a la misma ontología, pero están en archivos separados, y al mismo tiempo no todas las ontologías poseen instancias, por lo que estos ficheros en lenguaje RDF pueden existir o no.

Para solucionar este problema, después de cargar la ontología, se optó por comprobar si existe algún archivo más, aparte de la propia ontología, en el directorio actual. En el caso de que exista algún fichero con extensión RDF, se supondrá que contiene instancias, por lo que serán leídas y añadidas al modelo de la ontología que se acaba de cargar.

11) Se constató la utilidad de mostrar las anotaciones en una tabla. Ello es debido a que, aunque se muestran las anotaciones en el documento mediante el resaltado del texto con el color del concepto unidad, puede existir la duda sobre a qué concepto pertenece. Esta duda surge porque: las anotaciones tienen colores semejantes; una anotación es ocultada por otra que comprende el texto de la primera; o se quiere saber los atributos y valores de una anotación o las anotaciones que se relacionan con una determinada.

    La mejor forma de solucionar este problema consistía en la utilización de dos tablas: en una se mostrarían las anotaciones con sus atributos y valores, y en la otra las relaciones entre unidades.

    La primera de las tablas mencionadas, la de unidades, atributos y valores tendría como columnas: el texto de la anotación, el concepto unidad, el atributo al que se dará un valor, y la última columna indicará el valor asignado a este atributo. Como la misma anotación podría tener un mismo atributo con valores distintos, existirá una fila por cada valor asignada a cada atributo para cada unidad anotada.

    La segunda de las tablas, la de relaciones, tendría como columnas: el texto de la anotación origen, el nombre del concepto de la anotación origen, el texto de la anotación destino, el nombre del concepto de la

anotación destino, y por último, el nombre de la relación entre ambas anotaciones.

Además, para facilitar la identificación de las anotaciones en el texto, se posibilitó que, cada vez que se pulse una anotación en el texto, se resalten todas las filas de las dos tablas en las que la anotación tome parte.

12) Dada una selección del texto, en la herramienta original, no se le podría asociar más que una anotación, pero puede darse el caso de querer anotar el mismo texto con otra unidad, formando una nueva anotación. El objetivo es poder tener más de una anotación para el mismo texto, y luego poder seleccionar una u otra a pesar de quedar solapadas en la pantalla de visualización.

    El no permitir más de una anotación sobre el mismo texto provenía de una restricción que tenía la herramienta. Antes de crear una nueva anotación, se comprobaba si existía ya una anotación con las mismas posiciones de comienzo y fin. En el caso de que existiese, no se permitía la creación de la nueva anotación. Para solucionarlo, se añadió una condición extra a esta restricción, para que se permita crear una anotación adicional. Aparte de tener que coincidir las posiciones de inicio y fin, debía coincidir el concepto unidad con el que se intenta crear la anotación. Así cuando se haga clic sobre una anotación del texto, aparecerá una lista de anotaciones, cuyo texto comprenda la zona sobre la que se realizó el clic. De esta forma quedó resuelto el problema de tener solapadas varias anotaciones.

13) Disociar atributos con su correspondiente valor de las anotaciones y eliminar relaciones entre anotaciones, eran nueva funciones que debían estar presentes, ya que el usuario se puede confundir o percatarse más tarde de que estas anotaciones no eran correctas o

necesarias, y de esta forma querer disociar el atributo con su valor o eliminar la relación.

Debido a que la única forma de acceder a través de la aplicación, a los atributos de las anotaciones y a sus relaciones es mediante su tabla correspondiente, se deberían disociar y eliminar estos a través de dichas tablas. En el caso de los atributos con su valor, se selecciona la fila que representa el atributo y el valor de la anotación y se pulsa el botón "Delete attribute-value". Y para el caso de las relaciones se selecciona, en la tabla de relaciones, la fila que representa la relación que se quiera eliminar, y se pulsa el botón "Delete relation".

14) En la herramienta final, se añadieron los cambios necesarios para que los atributos y los valores de las anotaciones y las relaciones entre estas, se puedan crear rellenando los campos de la parte inferior derecha. Para rellenar un campo, basta con pulsar sobre un concepto del árbol de ontologías: para los atributos, valores y relaciones, o seleccionar una anotación con el botón izquierdo del ratón, para la anotación a la que asociaré un atributo-valor, para la anotación origen y para la anotación destino. Pero también se pensó en la opción de manejar estos conceptos de ontologías y anotaciones mediante *drag and drop* (traducido al español: "arrastrar y pegar").

    Por lo tanto, se permitió arrastrar conceptos, desde el árbol que representa la ontología, hasta las tablas de unidades y relaciones, y soltarlos en la fila adecuada. Los conceptos con los que se permite interactuar mediante *drag and drop* son los pertenecientes a las ontologías de atributos, valores y relaciones.

    Para poder arrastrar las anotaciones desde el documento hasta la tabla de relaciones, previamente hay que desactivar el concepto

seleccionado mediante el botón “Unselect” y seleccionar la anotación en toda su extensión.

15) La herramienta resultante se encuentra disponible en el lugar http://www.xp-dev.com/. Se puede acceder en modo lectura con el usuario “OntoLingAnnotizerLector” y la contraseña “Lectura”, ambos sin las comillas.

CONCLUSIONES

Llegado a este punto, se expondrán las conclusiones a las que se ha llegado, primero acerca del contexto en el que se encuadra la herramienta, y después de la propia herramienta de anotación OntoLing Annotizer.

La Web ha crecido sobremanera, así que tener acceso a la información, buscarla, organizarla o distribuirla se torna cada vez más difícil. Una de las posibles soluciones, cada vez más cercana, es extender la web actual y convertirla en la Web Semántica. La base en la que se fundamenta esta nueva web es que las máquinas sean capaces de comprender la semántica de los recursos web, lo que abrirá un amplio abanico de nuevas posibilidades. A mi parecer, llegar a tener una Web Semántica totalmente funcional sería un gran avance, y se podrían hacer cosas que hoy parecen de ciencia ficción. Pero también me parece que todo el proceso hasta llegar a la Web Semántica es muy largo y complicado; además una vez alcanzado el objetivo, la elaboración de recursos para esta nueva web sería bastante laborioso, y solamente un reducido grupo de personas serían capaces de crear estos documentos, mientras que en la actualidad casi cualquier persona es capaz de ello. El proceso lo considero largo y complejo porque se necesita un gran consenso sobre cómo y con qué anotar los documentos. Ante la dificultad de anotar los documentos, hay que reconocer que se están produciendo avances, intentando que sea más fácil, e incluso intentando llegar a un nivel en el que la anotación sea semiautomática o automática.

Respecto a la herramienta OntoLing Annotizer, se han cubierto todos los objetivos planteados al iniciar el proyecto, por lo que se puede afirmar que se dispone de una herramienta de anotación totalmente operativa para el modelo ontológico para el que fue diseñada. La interfaz gráfica es bastante intuitiva y sencilla, pero permite realizar todas las operaciones necesarias para una correcta anotación. La sencillez, en parte, es debida al modelo de anotación, donde las unidades tienen atributos que toman valores, y se

pueden relacionar con otras unidades. Todos los conceptos de las ontologías con las que se anota tienen un color distinto, y es con el que quedan resaltadas las anotaciones realizadas, por lo que se pueden visualizar las anotaciones en el texto y diferenciar con frecuencia a simple vista a qué concepto de la ontología de unidades pertenece cada anotación. A pesar de todo, no siempre se puede conocer a simple vista a que concepto pertenece una anotación por el color, ya que puede haber conceptos con colores muy parecidos. Para solucionar esto, y saber de qué concepto se trata se puede consultar la anotación en la tabla de anotaciones correspondiente: de unidades, atributos y valores, o de relaciones. Otras formas de identificar el concepto unidad son: hacer clic con el botón central, tras lo cual aparecerá un menú contextual que indicará el texto de la anotación y el concepto unidad; y hacer clic sobre ella con el botón principal o secundario, apareciendo una lista con texto de las anotaciones y el nombre del concepto, que recaen en esa zona.

Si bien la herramienta es bastante completa y sencilla, una vez terminada, se ha analizado para identificar posibles mejoras con las que extender OntoLing Annotizer, como: restringir el uso de atributos, valores y relaciones, en función de la unidad a la que pertenezca la anotación; hacer uso de la inteligencia artificial para conseguir un proceso semiautomático o automático; y en tercer lugar, controlar los niveles de anotación.

TRABAJOS FUTUROS

En este apartado se suguieren futuras líneas de trabajo para, de esta forma continuar el desarrollo de la actual herramienta y ampliarla, dotándola de nuevas funciones o mejorar las existentes. Estas líneas de trabajo se detallan en los puntos siguientes:

- Una posible ampliación de la herramienta sería poder restringir los atributos que pueden tener las anotaciones, los valores que pueden tomar estos atributos, con qué relaciones se asocian y cuáles son los objetos destino de estas relaciones. Todas estas restricciones estarían descritas de forma explícita en una ontología, y bastaría con recorrer esta ontología para comprobar si es viable una acción de las arriba descritas. Para determinar si una anotación puede tener un atributo o no, habría que comprobar a qué concepto de la ontología de unidades pertenece la anotación y verificar si este concepto puede poseer este atributo. En el caso de que lo pueda poseer, posteriormente, cuando se le vaya a asignar a este atributo un valor, se comprobaría también si el valor está dentro del dominio de valores que puede tomar el atributo. Determinar qué relaciones pueden usarse con que unidades requeriría un proceso similar.
- Dotar de Inteligencia Artificial (IA) a la herramienta para facilitar el proceso de anotación al usuario. Esta mejora sería, en principio más complicada, ya que requiere de técnicas de IA, lo que de por sí podría ser objeto de un estudio extenso.
- OntoLing Annotizer se desarrolló según un modelo de ontologías ya diseñado y realizado, o a punto de concluir. Este modelo ontológico para la anotación semántica se basa en la anotación de corpus de la Lingüística de Corpus. En dicho modelo de anotación lingüística existen distintos niveles de anotación, perteneciendo a cada uno unidades, valores y relaciones distintos. Estos niveles son: anotación

de lemas, anotación morfosintáctica, anotación sintáctica, anotación semántica y anotación del discurso. En OntoLing Annotizer se encuentran todos los conceptos distribuidos entre cuatro distintas ontologías, pero no separadas por niveles. Otra posible ampliación sería decidir en cada momento en qué nivel se está anotando y, consecuentemente, mostrar únicamente los atributos pertenecientes a ese nivel.

REFERENCIAS BIBLIOGRÁFICAS

- Aguado, G., Álvarez-de-Mon, I. y Pareja, A. (2002a). Primeras aproximaciones a la anotación lingüístico-ontológica de documentos de la Web Semántica: OntoTag. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.17, pp. 37–49.
- Aguado, G., Álvarez-de-Mon I., Gómez-Pérez, A., Pareja-Lora, A., Plaza-Arteche, R. (2002b). "A Semantic Web Page Linguistic Annotation Model". *Semantic Web Meets Language Resources. Technical Report WS-02-16*. AAAI Press. Menlo Park, California, E.E.U.U.
- Aguado, G., Álvarez-de-Mon, I., Gómez-Pérez, A. y Pareja, A. OntoTag's Linguistic Ontologies: Improving Semantic Web Annotations for a Better Language Understanding in Machines (2004). Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04). IEEE Computer Society.
- Benjamins, V.R., Fensel, D., Decker, S. y Gómez-Pérez, A. (1999) $(KA)^2$: Building Ontologies for the Internet: a Mid Term Report. IJHCS, International Journal of Human Computer Studies, 51: 687–712.
- Benjamins, V.R., Contreras, J., Blázquez, M., Dodero, J.M., García, A., Navas, E., Hernández, F. and Wert, C. (2004). 'Cultural heritage and the semantic web', in Bussler, C., Davies, J., Fensel, D. and Studer, R. (Eds.): *The Semantic Web: Research and Applications*, *First European Semantic Web Symposium (ESWS2004)*, Springer-Verlag, pp.433–444.
- Berners-Lee, T., Hendler, J., Lassila, O. (2001). *The Semantic Web*. Scientific American.
- Castellanos, D. (2007). Sistema basado en tecnologías de la web semántica para evaluación en entornos de e-learning. Universidad de Murcia. Tesis doctoral accesible mediante la dirección: http://www.tesisenred.net/TESIS_UM/AVAILABLE/TDR-0922108-131211//CastellanosNieves.pdf.

- Ciravegna, F. (2001). 'Adaptive information extraction from text by rule induction and generalisation', in Nebel, B. (Ed.): *17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, Seattle, ashington, Morgan Kauffmann Publishers, San Francisco, California, pp.1251–1256.
- Corcho, O. (2006). 'Ontology based document annotation: trends and open research problems'. *Int. J. Metadata, Semantics and Ontologies, Vol. 1, No. 1*
- Gómez Pérez, A., Corcho, O., Fernandez-Perez, M., (2003). *Ontological Engineering*. Springer Verlag.
- Gruber, T. R. (1993). "A Translation Approach to Portable Ontologies". Knowledge Acquisition, 5(2): 199-220.
- Heijst, V., Schreiber, G., A.T., Wielinga, B. J. (1997). "Using explicit ontologies in KBS development". *International Journal of Human Computer Studies*, 45, 183-292.
- ISO (International Organization for Standardization). *ISO 8879:1986(E). Information processing — Text and Office Systems — Standard Generalized Markup Language (SGML).* First edition — 1986-10-15. [Geneva]: International Organization for Standardization, 1986.
- Lacy, Lee W. (2005). *OWL : representing information using the web ontology language*. Victoria : Trafford
- Luke S. y Heflin J. (2000) SHOE 1.01. Proposed Specification. SHOE Project. http://www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm
- Mizoguchi, R., Vanwelkenhuysen, J. e Ikeda, M. (1995). "Task Ontology for Reuse of Problem Solving Knowledge. Towards Very Large Knowledge Bases". KnowledgeBuilding *and Knowledge Sharing*, 46-59.
- Motta, E., Buckingham Shum, S. y Domingue, J. (1999) Case Studies in Ontology-Driven Document Enrichment. Proceedings of the 12th Banff Knowledge Acquisition Workshop, Banff, Alberta, Canada.

- Pareja, A. (2007). Representación y recuperación automatizada en bibliotecas y servicios documentales. Web Semántica y ontologías. http://www.fesabid.org/santiago2007/descargas/mesas/apareja_lora.pdf.
- Poli, R. (2000). "Levels of Reality. BISCA 2000: Bolzano International Schools". In *Cognitive Analysis Dependence and Dynamic Categories*.
- Poli, R. (2001). Alwis: Ontology for Knowledge Engineers, PhD thesis, Utrecht.
- Poli, R. (2002). *Descriptive, Formal, and Formalized Ontologies*. In D. Fisette (ed.), Edmund Husserl''s Logical Investigations 1901–2001: Origins and Posterity of Phenomenology (forthcoming).
- Russell, S. y Norving, P. (2004). Inteligencia artificial: un enfoque moderno. Madrid : Prentice Hall.
- Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Mädche, A., Schnurr, H.-P. y Studer, R. (2000) Semantic Community Web Portals. WWW9 - Proceedings of the 9th International World Wide Web Conference, 33(1-6): 473-491 (Special Issue). Amsterdam, Holanda: Elsevier.
- Vossen, G. y Hagemann, S. (2007). Unleashing Web 2.0: From Concepts to Creativity. Morgan Kaufmann
- W3C. A Little History of the World Wide Web. http://www.w3.org/History.html [Consultado el 30-7-09].

DIRECCIONES DE INTERNET

- http://annotation.semanticweb.org/ontomat/index.html
- http://infolab.stanford.edu/OntoAgents/
- http://kmi.open.ac.uk/projects/akt/MnM/index.html
- http://protege.stanford.edu/
- http://www.aktors.org/
- http://www.dcs.shef.ac.uk/~ajay/html/cresearch.html
- http://www.drools.org/
- http://www.isoco.com/
- http://www.sun.com/download/
- http://www.sureshotsoftware.com/cavaj/index.html
- http://www.w3.org
- http://www.w3.org/2007/09/OWL-Overview-es.html
- http://www.w3.org/History.html#refs
- http://www.w3.org/RDF/
- http://www.w3.org/TR/owl-features/
- http://www.w3.org/TR/rdf-primer/
- http://www.w3.org/TR/rdf-schema/
- http://www.w3.org/TR/REC-xml-names/
- http://www.w3.org/XML/

APÉNDICE: MANUAL DE USUARIO

Este Apéndice contiene toda la información necesaria para que el usuario esté en disposición de ejecutar la herramienta, y para que, posteriormente, sea capaz de usar todas las funciones de las que dispone OntoLing Annotizer.

El primer subapartado contiene los requisitos necesarios para poder usar la herramienta, sin los cuales no se puede ejecutar. El segundo subapartado contiene las instrucciones para la ejecución de la herramienta, una vez cubiertos los requisitos. El último apartado describe las funciones que proporciona OntoLing Annotizer, describiéndolas paso a paso.

REQUISITOS

Para poder ejecutar la herramienta *OntoLing Annotizer* se debe tener instalada la JRE (Java Runtime Environment) 5, o alguna versión posterior. No se ha probado con versiones anteriores a la cinco, así que no se asegura su funcionamiento con versiones más antiguas. En caso de que el sistema no contenga la JRE, esta se puede descargar desde el sitio web de SUN, http://www.sun.com/download/.

En cuanto a la plataforma de ejecución no existe ninguna restricción asociada, por lo que se puede considerar una aplicación multiplataforma. Que sea multiplataforma se debe a la utilización de Java como lenguaje para su desarrollo.

Respecto a permisos de lectura-escritura, se debe tener permisos de lectura y escritura en la carpeta de ejecución. Esto es debido a que, al ejecutarse por primera vez, se debe crear la estructura de ficheros necesaria para su funcionamiento; asimismo, cada vez que se guarde el estado del

documento activo, se escribirá el archivo de salida con los cambios realizados en el documento actual y en la sesión abierta.

EJECUCIÓN

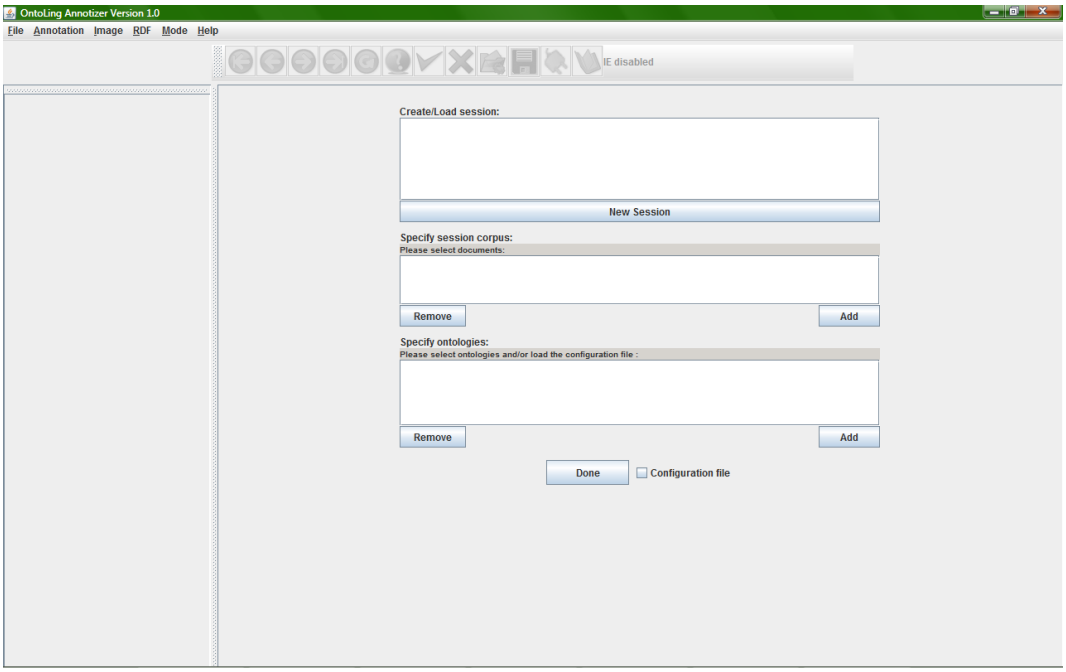
La herramienta se distribuye como un archivo ejecutable comprimido de Java, archivos con extensión "jar", junto a una carpeta, con el nombre *lib*, que contiene todas las librerías que son necesarias para su ejecución. Para ejecutar OntoLing Annotizer, tanto el ejecutable Java como la carpeta con las librerías deben estar en la misma ubicación. Aparte de esto, no se requiere de ningún otro proceso de instalación o configuración. Iniciar la herramienta es tan sencillo como lanzar el ejecutable. Tras ello, se crearán la estructura de carpetas y los archivos necesarios para su funcionamiento. Estos archivos serán creados en la misma carpeta en la que se encuentra el ejecutable.

FUNCIONES

En este apartado se mostrarán todas las funciones que posee la herramienta, explicándolas de forma detallada y acompañadas de imágenes de esta para su mejor entendimiento.

Nada más ejecutar la herramienta, aparece la pantalla principal, donde se selecciona o se crea un nuevo un usuario, se decide la composición del corpus y se escogen el conjunto de ontologías con las que se anotará el misimo. Esta pantalla principal se puede observar en la **Figura 2**.

Figura 2: Pantalla principal de la aplicación OntoLing Annotizer.



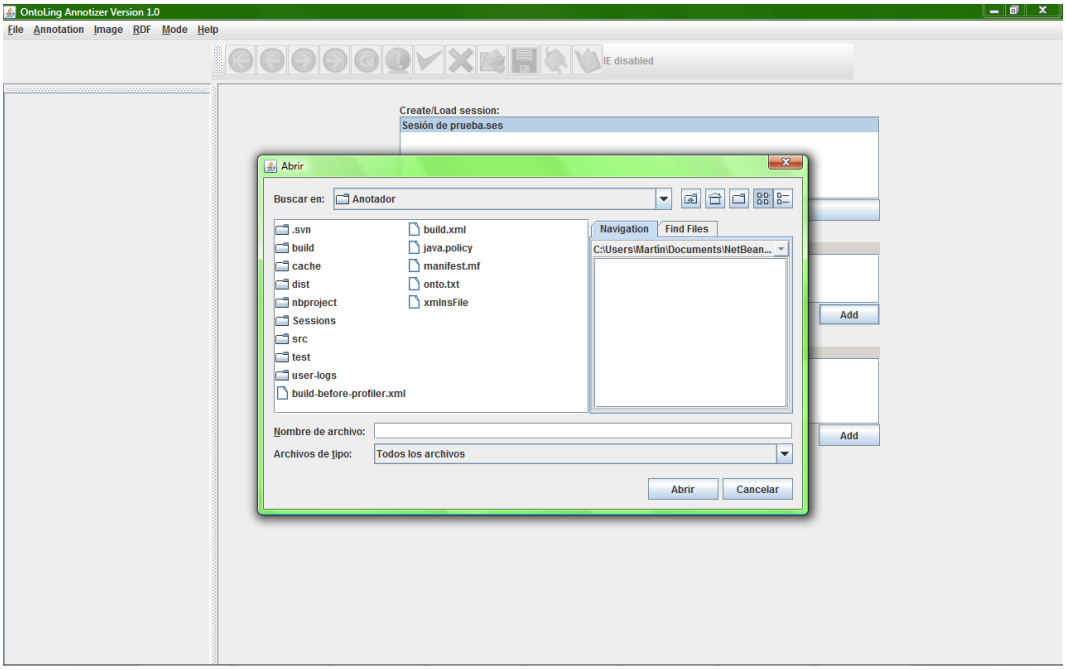
Debajo de la etiqueta de texto "Create/Load session:" se encuentra un panel de texto que contiene las sesiones disponibles, en caso de que exista alguna. En la **Figura 2**, el panel de sesiones no contiene ninguna, debido a que es la primera ejecución de la herramienta. Para crear una nueva sesión, se debe presionar el botón "New Session". Aparecerá entonces una ventana solicitando un nombre de sesión. Una vez se haya escrito un nombre para la nueva sesión, se pulsa el botón *Aceptar* y, a continuación, aparecerá la nueva

sesión como disponible. Pulsar el botón *Cancelar* no producirá ningún efecto, por lo que no se creará la nueva sesión. Cabe observar que, a la hora de introducir el nombre de sesión, que se aceptará cualquier cadena de texto que se pueda teclear (espacios, guiones, letras acentuadas, etc.).

SELECCIONAR EL CORPUS

Una vez se haya seleccionado la sesión con la que se desea continuar utilizando la herramienta, se debe constituir el corpus que se va a anotar. Para seleccionar los documentos que formarán el corpus, se debe pulsar el botón *Add* que se encuentra debajo de panel de texto con la etiqueta "Specify session corpus:" (véase **Figura 2**), entonces aparecerá un selector de ficheros, como se muestra en la **Figura 3**, cuyo directorio actual será el directorio padre del directorio que contiene el ejecutable. En este selector de ficheros se debe escoger el archivo o los archivos con extensión HTML con los que se quiere formar el corpus. En el caso de añadir los documentos de uno en uno, basta con seleccionar el documento deseado y pulsar el botón *Abrir*. Automáticamente, el archivo pasará a formar parte del corpus, volviendo a la pantalla principal. Si se quiere añadir más de un documento (que se encuentre en la misma carpeta) basta con pulsar la tecla *control* del teclado cada vez que se haga clic sobre un nuevo archivo. Una vez estén seleccionados todos los archivos que se quieren añadir, se pulsa el botón *Abrir*, volviendo de nuevo a la pantalla principal. Si se quieren añadir más archivos pertenecientes a otra carpeta, o simplemente se ha olvidado alguno de la misma carpeta, basta con repetir cualquiera de las acciones anteriores, la de añadir los archivos de uno en uno o haciendo uso de la tecla *control*.

Figura 3: Selector de ficheros desde el que se escoge el corpus.



Existe una tercera posibilidad de seleccionar los documentos que constituirán el corpus. En muchas ocasiones, los documentos del corpus estarán en una misma carpeta del sistema de ficheros. Esta agrupación tiene mucha lógica, ya que forman una unidad. Debido a esto, la tercera posibilidad permite que, en el selector de ficheros de la **Figura 3**, se pueda seleccionar la carpeta que contenga los documentos del corpus, en vez de ir seleccionando estos archivos de uno en uno. Una restricción que tiene esta última forma de selección del corpus es que la carpeta puede contener exclusivamente los archivos que constituyen el corpus, todos con extensión HTML, sin incluir ningún archivo más.

SELECCIONAR LAS ONTOLOGÍAS

El proceso para indicar las ontologías con las que se procederá a anotar es análogo al proceso para indicar el corpus. Al igual que con el caso del corpus, existe un panel con las ontologías existentes en la sesión actual. Este panel se puede ver en la **Figura 2**: es el que se encuentra justamente debajo de la etiqueta "Specify ontologies:". Pulsando el botón *Add* bajo este panel, se despliega un selector de ficheros, semejante al que se mostraba al presionar el botón *Add* superior. Con este selector de ficheros se seleccionan las ontologías. También aquí, de nuevo, se tienen las tres formas de seleccionar los documentos que constituyen (en este caso) las ontologías. Estas tres formas son: añadir las ontologías de una en una; añadir un conjunto de ontologías almacenadas dentro de la misma carpeta, seleccionándolas de una en una; y por, último, seleccionar una carpeta que contenga ya las ontologías, añadiéndose de esta forma todas las ontologías que existan dentro de esa carpeta de una vez.

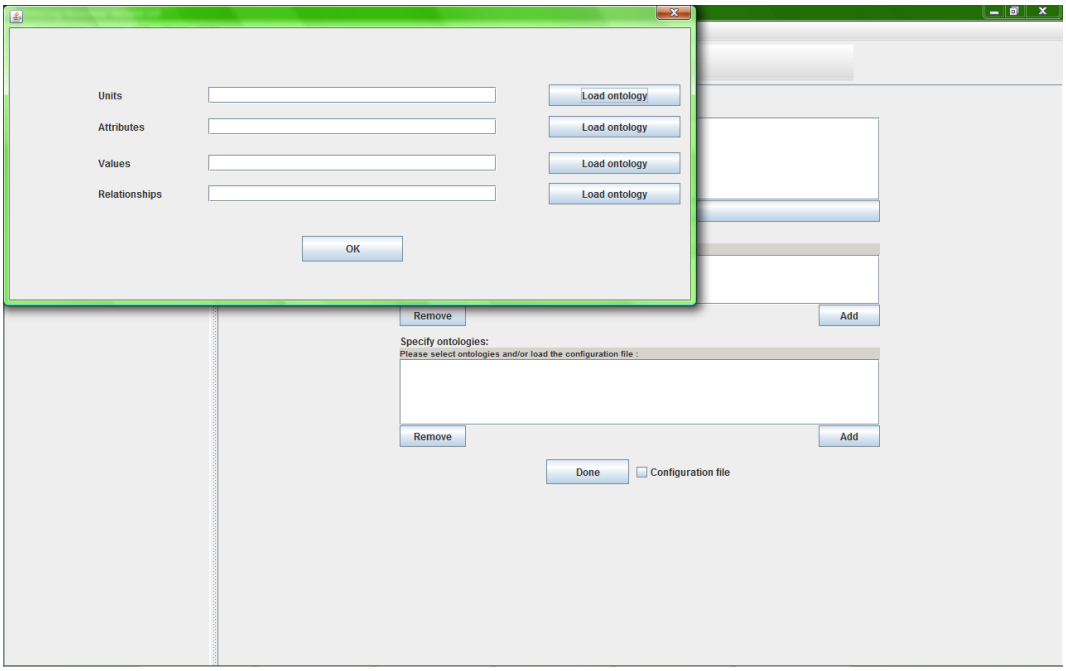
Además de las tres formas anteriores de seleccionar las ontologías existen otras dos. La primera de ellas se deriva de la estructura asociada a las ontologías. Como ya se explicó anteriormente, la herramienta se ha desarrollado pensando en que se necesitan, al menos, cuatro ontologías. Estas cuatro ontologías son: la de unidades, la de atributos, la de valores y la de relaciones. Para utilizar la herramienta de manera coherente, con el objetivo para el que se ha desarrollado la aplicación, se requieren estas cuatro ontologías. Por este motivo, la cuarta opción consiste en activar un *check box* con el que se cargan automáticamente las ontologías que se indican en un archivo de configuración. Este *check box* se encuentra en la parte inferior de la pantalla que se muestra en la **Figura 2** y se denomina "Configuration file". Si se activa este *check box* no se actualiza el panel de las ontologías, pero cuando se inicie el proceso de anotación se dispondrá de las ontologías indicadas en el archivo de configuración. Este archivo de

configuración puede modificarse mediante un formulario al que se accede mediante el menú superior de la pantalla principal (véase la **Figura 2**).

La quinta y última forma de añadir una ontología se concreta a través de los menús superiores, una vez se haya abandonado la pantalla inicial y se esté en situación de comenzar a realizar anotaciones. Dentro del menú "File" (**Figura 2**), existe la opción "Load ontology", con la que se da paso a un selector de ficheros, con el que se puede añadir una ontología. Más adelante se darán indicaciones de cómo cargar ontologías de forma dinámica, una vez se haya iniciado el proceso de anotación.

El acceso al archivo de configuración, ya sea para comprobar o para modificar las ontologías que se cargarán, se realiza mediante un formulario. A dicho formulario se llega a través del elemento “Configuration file” del menú “File”.

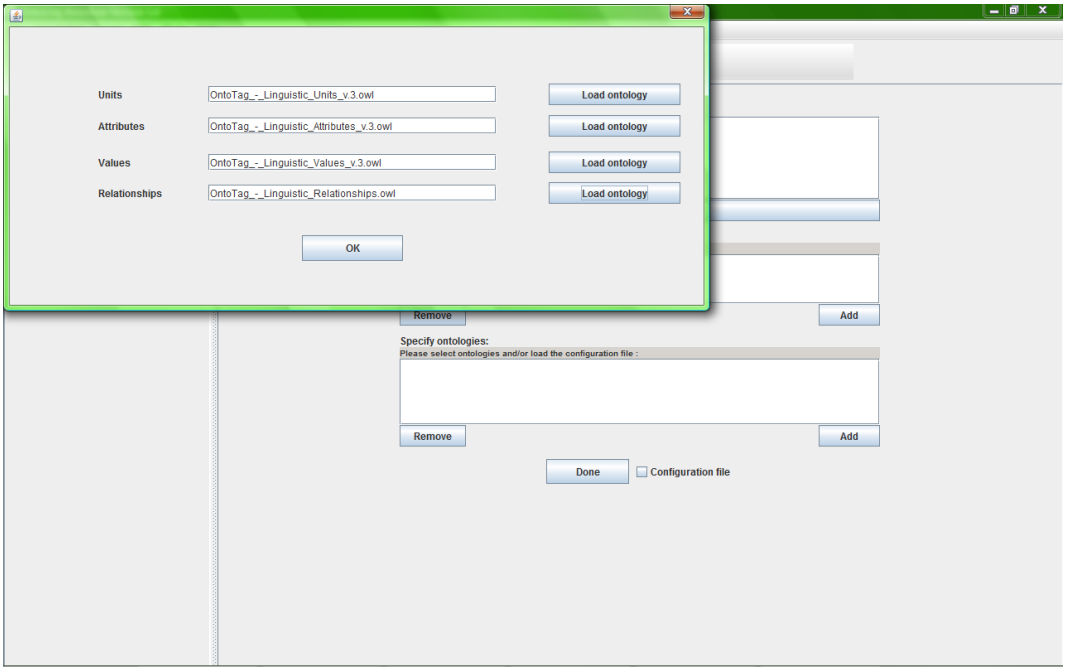
Figura 4: Formulario para la modificación del archivo de configuración de ontologías.



Como puede verse en la **Figura 4**, en este formulario existen cuatro botones y cuatro campos. Cada uno de estos botones y campos está asociado a un tipo de ontología: Units, Attributes, Values y Relationships. Se puede saber qué botón y qué campo están relacionados con qué tipo de ontología, gracias a las etiquetas que se encuentran a su izquierda. En los campos aparecen los nombres de los archivos, en formato OWL, que describen la ontología. Los botones tienen la finalidad de cambiar las ontologías que se utilizarán en el

proceso de anotación. Presionando uno de los botones mencionados, aparece un selector de ficheros con el que localizar la ontología que se desea utilizar. Una vez se hayan seleccionado todas las ontologías, se mostrarán sus nombres en los campos, como se observa en la **Figura 5**, pero falta validar el proceso para que sean cargadas. La validación se produce presionando el botón *OK*. Si no se pulsa este botón y se cierra la ventana, los cambios no son guardados. Por el contrario, al hacer clic sobre el botón *OK,* los cambios son guardados y las referencias a los ficheros que describen las ontologías son almacenados, cerrándose la ventana y volviendo a la pantalla principal.

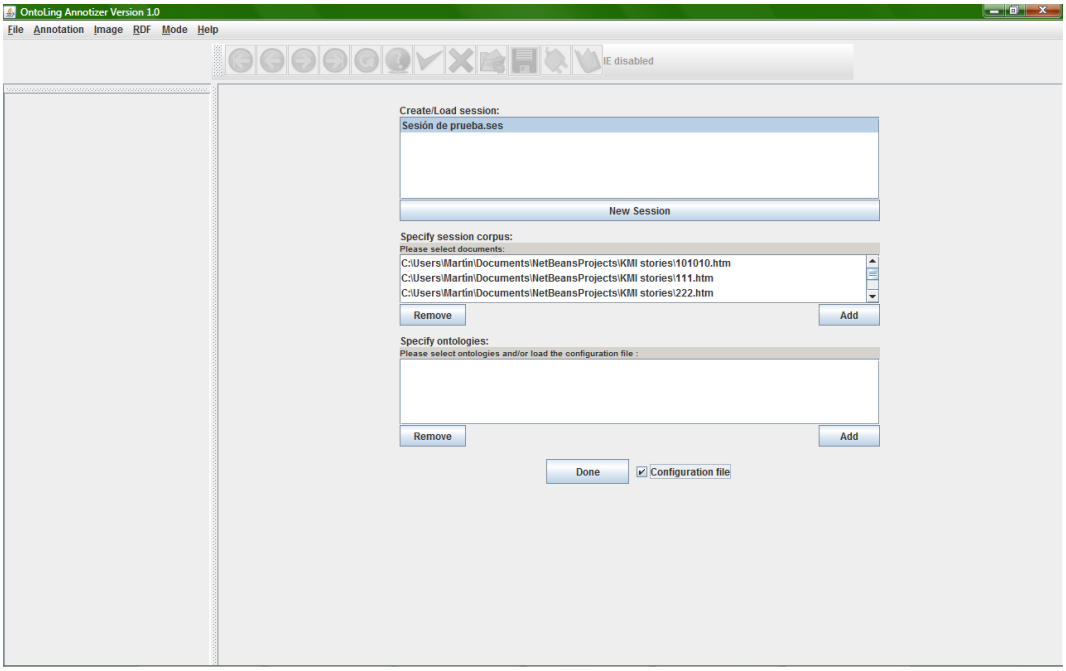
Figura 5: Formulario con todas las ontologías seleccionadas.



Una vez cumplidos los requisitos: haber escogido una sesión, a la que se ha añadido un corpus y un conjunto de ontologías (al menos una, ya sea mediante el *check box* o sin él) se puede iniciar el proceso de anotación. Permaneciendo en la pantalla principal y cumplidos los requisitos mencionados, basta con hacer clic en el botón con el texto *Done* para que comience la carga de la sesión. Si no se ha seleccionado una sesión, si el corpus no tiene ningún documento, o si no se ha escogido ninguna ontología (activar el "Configuration file" es equivalente a seleccionar cuatro ontologías), se mostrará un mensaje de error, indicando a cuál de estas tres causas es debido.

En la **Figura 6** se puede observar el resultado tras seleccionar una sesión, y un conjunto de documentos que pertenece al corpus. Como también está activado el *check box* del archivo de configuración, basta con presionar el botón *Done* para iniciar la sesión.

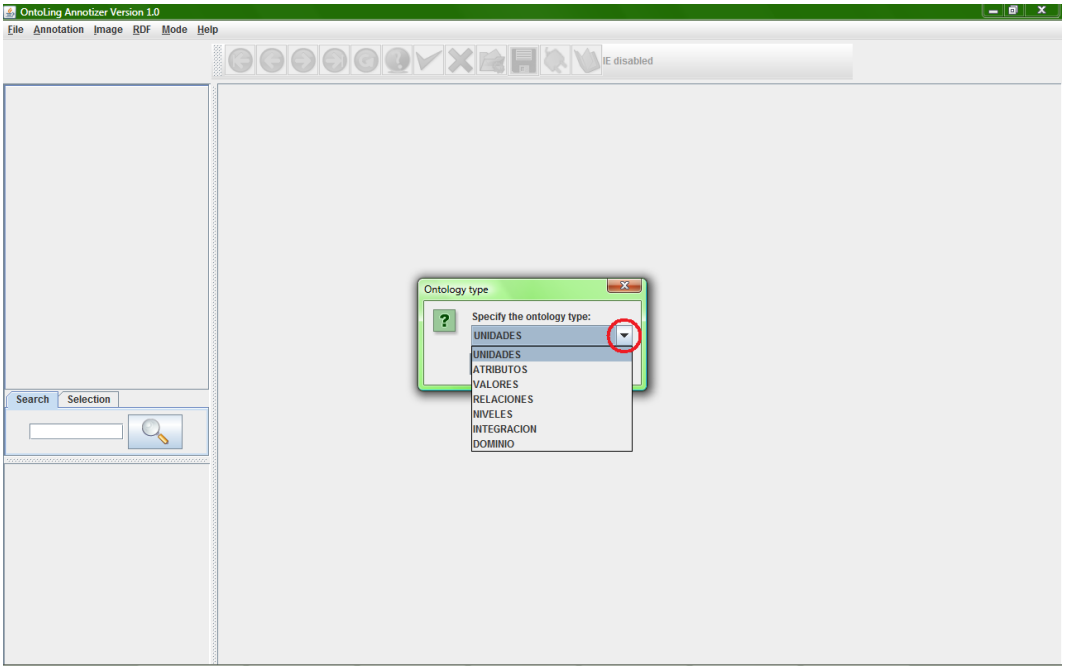
Figura 6: Pantalla principal con los requisitos cumplidos para iniciar la sesión.



Nada más comenzar la carga se solicitará un nombre por cada ontología seleccionada, para identificarla. Este nombre no cambiará el que tenga el archivo, ni su contenido, sino que servirá para identificar cada una de las ontologías cuando se esté anotando. En el momento en el que se solicita el nombre, se muestra el nombre del archivo y el de la carpeta que lo contiene, para saber a qué ontología se refiere. Por defecto se sugiere, como nombre para la ontología, el que tiene el archivo que la contiene. Este nombre identificativo se puede volver a cambiar, tantas veces como se quiera, una vez se haya comenzado el proceso de anotación. Algunos identificadores, incluso todos, pueden ser exactamente iguales. Aunque no se obliga a que sean únicos para cada ontología, se recomienda que sean distintos, con el objetivo de poderlas diferenciar perfectamente sin recurrir a su contenido.

Dependiendo de si la ontología se ha cargado mediante el botón *Add*, o mediante el archivo de configuración, se mostrará o no, respectivamente, una lista desplegable (**Figura 7**), donde el usuario tendrá que indicar de qué tipo es la ontología. Para mostrar los tipos de ontologías que se pueden seleccionar, basta hacer clic sobre la zona que abarca el círculo rojo de la **Figura 7**. Si la ontología se ha cargado con el archivo de configuración, no se solicitará el tipo, porque ya se le asoció uno en aquel momento.

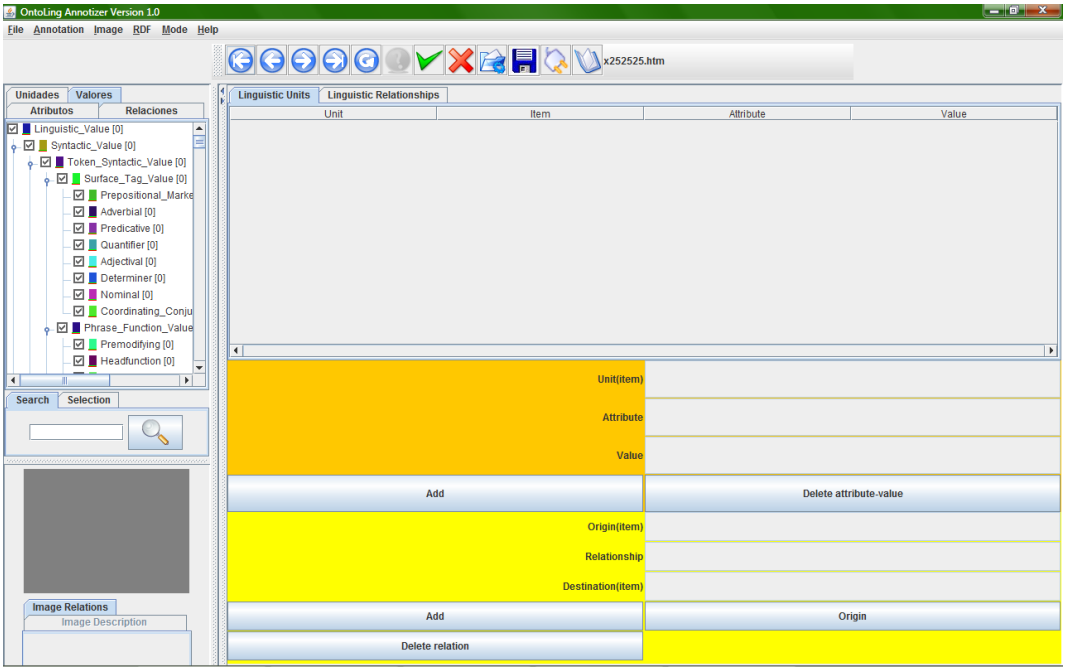
Figura 7: Lista desplegable con los tipos de ontologías.



Una vez se hayan indicado los nombres de las ontologías y sus tipos, (estos últimos sólo cuando sea necesario), se presentará la pantalla de anotación de la **Figura 8**. Puede suceder que no se visualice exactamente la misma pantalla: esto depende de cuál haya sido la última ontología en cargarse, y por lo tanto, la que esté activa. Cuando está activa una ontología de unidades, se muestra el texto que se procederá a anotar, como en la **Figura 8**. Sin embargo, si la ontología activa no es de unidades, no se mostrará el texto, como ocurre en la **Figura 9**.

Figura 8: Pantalla de anotación con una ontología de unidades activa.

Figura 9: Pantalla de anotación con una ontología activa distinta de la de unidades.



En el momento de realizar una anotación debe estar seleccionada una ontología de unidades, ya que son estas las que se anotan, que pueden tener unos atributos, los cuales toman unos valores, y pueden estar relacionadas con otras unidades mediante relaciones. Activar una u otra ontología de las cargadas, es tan simple como seleccionar alguna de las pestañas que se encuentra en el panel izquierdo. Podemos diferenciar las ontologías gracias a los identificadores que les asignamos. Como se introdujo en su momento, se pueden cambiar los nombres que poseen las pestañas. Al realizar doble clic sobre el nombre de la pestaña, se muestra una ventana donde puede introducirse el nuevo nombre que se le va a dar a la pestaña.

Tras activar la ontología necesaria, se debe buscar en la estructura arbórea la instancia del concepto con el que se anotará. Cada estructura arbórea es una representación de una ontología. Estas representaciones no tienen necesariamente una raíz única, a no ser que exista un superconcepto del que hereden todos los demás. Encontrado el concepto, se debe seleccionar. Para ello basta con hacer clic sobre él. Para comprobar que ha sido seleccionado, se puede consultar el campo de la pestaña “Selection”, que se encuentra justamente debajo de la ontología. En la **Figura 10** se puede comprobar que está activa la ontología “Unidades” y se ha seleccionado el concepto “Sentence”, pues en el campo de la pestaña “Selection” se muestra el nombre del concepto. En la **Figura 10**, se ha rodeado con un círculo rojo el campo, para facilitar su localización.

Figura 10: Pantalla de anotación con el concepto "Sentence" seleccionado

Tras seleccionar un concepto, ya se puede realizar una anotación. Para ello, se debe hacer clic con el botón izquierdo sobre la zona del texto donde comenzará la anotación y, sin soltar el botón, arrastrar hasta donde se quiere que termine la anotación. Como las palabras, junto con los signos de puntuación, suelen ser los elementos básicos en la anotación, una anotación normalmente no tomará una parte de una palabra, por lo que, aunque se comience o termine la selección en la mitad de una palabra, la selección se extenderá hasta que abarque toda la palabra. En cuanto se suelte el botón, la fase de selección terminará y se tendrá una nueva anotación, quedando resaltado el texto con el mismo color que tiene el concepto en el árbol de la izquierda. Además, la herramienta también buscará en el documento otras ocurrencias de los caracteres de la cadena anotada. Estas ocurrencias serán sugeridas para su anotación y aparecerán señalizadas con un rectángulo, cuyo borde será del mismo color que el concepto que aparece en la ontología, y con el fondo blanco. En caso de que se quieran aceptar todas las

sugerencias que propone la herramienta, basta con hacer clic sobre el botón con un tic verde y las anotaciones sugeridas serán confirmadas, tomando el fondo el mismo color que el borde. En caso de que se desee rechazar todas las sugerencias, se pulsará el botón cuyo icono es un aspa de color rojo. De esta forma desaparecerán los bordes y se producirá únicamente la anotación que realizó el usuario. En la **Figura 11** se muestra cómo el usuario realizó la anotación, de una ocurrencia de la palabra "the", y cómo la herramienta sugirió otras ocurrencias, también se muestran rodeados con círculos rojos los botones para rechazar y aceptar las sugerencias.

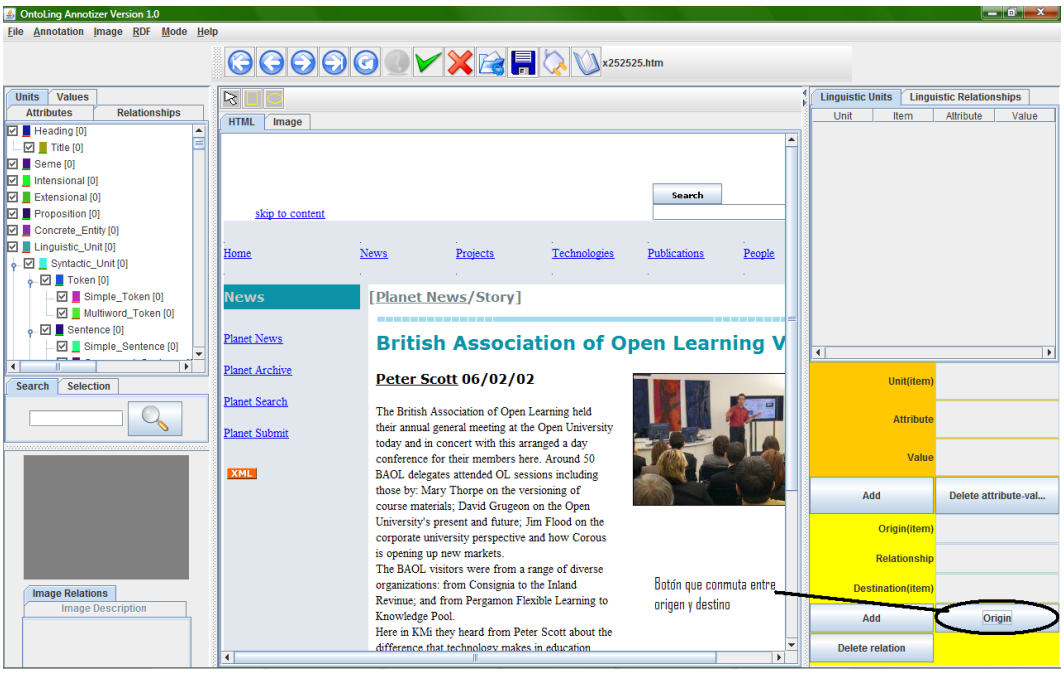
Figura 11: Botones para aceptar y rechazar las sugerencias de la herramienta.

Como se indicó anteriormente, las anotaciones no suelen tomar sólo una parte de una palabra, sino que, normalmente, la palabra entera participa en la anotación. Sin embargo, como pueden existir excepciones, se ha habilitado una forma de seleccionar una parte de una palabra sin

seleccionarla entera. Esto se consigue haciendo uso del teclado y el ratón. Para realizar una selección de este tipo, se pulsa el botón izquierdo del ratón y, sin soltarlo, se pulsa también la tecla *shift* del teclado. Posteriormente, sin dejar de soltar estos dos botones, el del ratón y el del teclado, se utilizan las teclas de dirección del teclado para acotar el alcance de la anotación. Una vez acotado su alcance la anotación se soltará el botón del ratón y se llevará a cabo la anotación.

Cuando se dispone de varias anotaciones en el texto, estas se pueden seleccionar realizando un clic con el botón principal del ratón sobre ellas. Tras esta operación, aparecerá una lista desplegable, donde se dará a escoger al usuario el conjunto de anotaciones cuyo texto comprende la palabra sobre la que se ha hecho clic. Para poder diferenciar bien las distintas anotaciones, se mostrará el nombre del concepto de la anotación y, entre paréntesis, el texto de esta. Seleccionando una de estas opciones, se rellenarán dos campos de la derecha: (1) "Unit (item)" y (2) "Origin (item)" o "Destination (item)". Que se complete uno u otro de estos dos últimos campos dependerá de si el botón que se encuentra a la derecha del botón *Add* inferior (**Figura 12**), muestra el texto "Origin" o "Destination", respectivamente. Para conmutar entre origen y destino basta con pulsar dicho botón.

Figura 12: Botón que conmuta entre origen y destino.



AÑADIR UN ATRIBUTO Y VALOR A UNA ANOTACIÓN

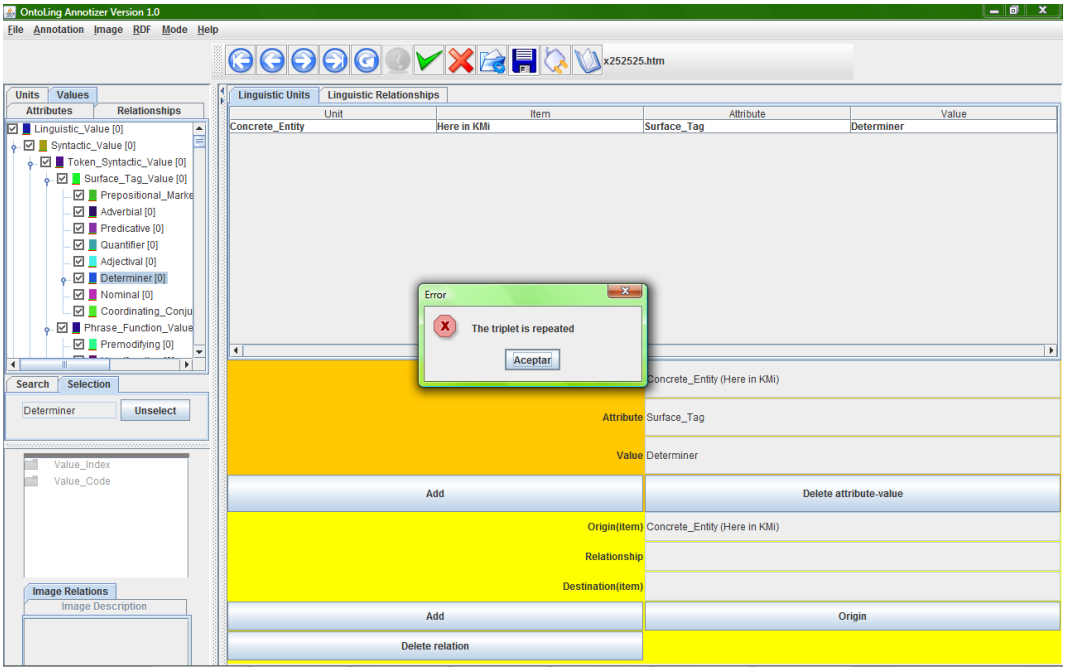
Existen tres formas distintas de asociar un atributo y un valor a una anotación: mediante el botón *Add* superior y los campos que se encuentran en el panel inferior derecho; mediante el botón secundario del ratón y arrastrando desde las ontologías a la tabla de atributos-valores; y realizando doble clic con el botón principal del ratón y arrastrando desde las ontologías a la tabla de atributos-valores.

Para añadir un atributo y/o un valor a una anotación utilizando el panel inferior derecho, primeramente hay que seleccionar una anotación, como se ha visto en la sección anterior. Teniendo ya, una anotación seleccionada se escoge el atributo o el valor de la ontología correspondiente. Si el concepto que se elige pertenece a una ontología de valores o atributos, automáticamente se rellenan los campos "Attribute" y "Value" de forma correcta. Es decir, si el concepto pertenece a una ontología de atributos se rellena el campo "Attribute"; y si pertenece a una de valores, se rellena el campo "Value". Una vez que se tienen rellenos los tres campos, "Unit", "Attribute" y "Value", se puede finalizar el proceso pulsando el botón *Add* superior, apareciendo una nueva fila en la tabla de unidades lingüísticas (en inglés, "Linguistic Units") con los datos introducidos. Si no aparece una nueva fila, significará que el proceso no ha finalizado con éxito, debido a que la tripleta Unidad-Atributo-Valor estaba ya en la tabla, y por lo tanto no se puede añadir.

Si se hace un clic con el botón secundario del ratón sobre una anotación, se desplegará una lista con la que escoger una anotación, sobre la que se seguirá actuando. Seleccionada una anotación, aparecerá un menú contextual (*pop-up menú*) con tres opciones: "Add Attribute/Value", "Add Relation" y "Remove Annotation". Se debe escoger la primera opción, "Add Attribute/Value". Esta elección provocará la apertura de una nueva entrada en la tabla de unidades lingüísticas, que no tendrá todos sus campos

ocupados. Únicamente estarán ocupados los dos primeros, “Unit” e “Item”. Para completar los dos restantes, “Attribute” y “Value”, basta con seleccionar la fila, en caso de que no esté seleccionada, y arrastrar algún concepto de una ontología de atributos para completar el campo “Attribute”, y de una ontología de valores para rellenar el campo “Value”. Cuando se ocupe la última celda sin un valor, se finalizará el proceso y se añadirá la nueva pareja de atributo-valor a la anotación, siempre y cuando no existiese ya. En caso de que la tripleta unidad-atributo-valor existiese ya, aparecerá un mensaje con el texto “The triplet is repeated”, no se asociará ni el atributo ni el valor a la anotación, y la fila que se estaba rellenando desaparecerá.

Figura 13: Mensaje indicando que la tripleta está repetida.



Para hacer uso de la tercera opción, debe estar visible la tabla de unidades lingüísticas, esto es, seleccionar la pestaña con este nombre. Cuando esté seleccionada esta pestaña, se realiza doble clic sobre una

anotación, desplegándose, como en los caso anteriores, una lista con las anotaciones disponibles. Escogiendo una anotación de la lista, se crea una nueva fila en la tabla de unidades lingüísticas. A partir de este momento, el proceso es idéntico al de la opción anterior, una vez creada la nueva fila.

RELACIONAR DOS ANOTACIONES

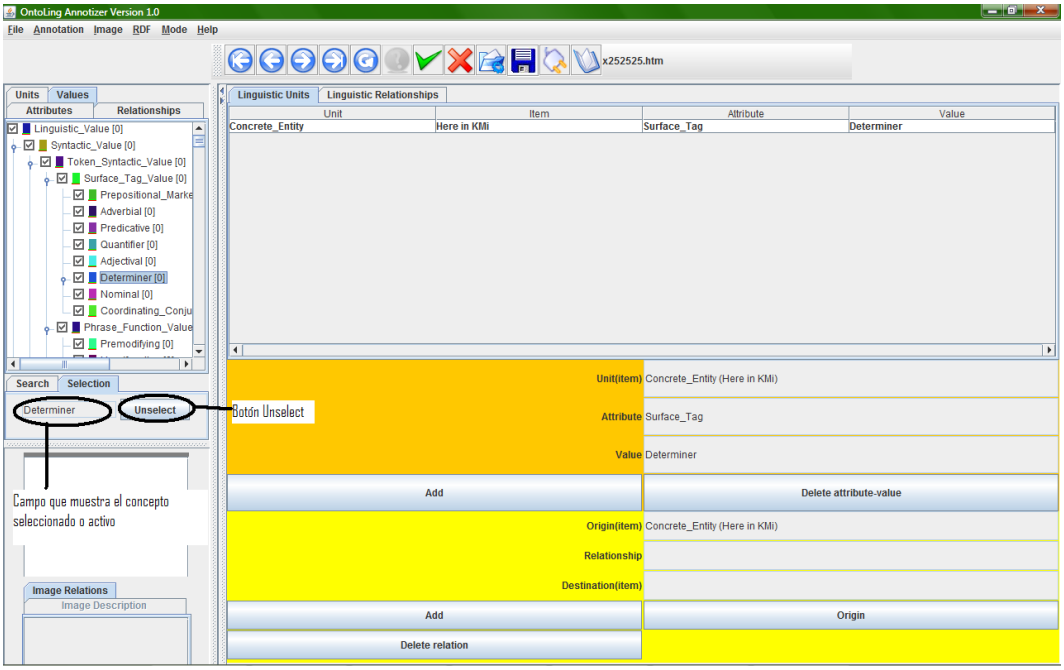
Existen tres métodos para relacionar dos anotaciones: utilizando el botón *Add* inferior y los campos que se encuentran en el panel inferior derecho; utilizando el botón secundario del ratón y arrastrando desde las ontologías y el documento HTML a la tabla "Linguistic Relationships" (en español, relaciones lingüísticas); y utilizando el doble clic con el botón principal del ratón y arrastrando desde las ontologías y el documento HTML a la tabla de relaciones lingüísticas. Estos tres métodos son semejantes a los tres utilizados para añadir un atributo y un valor a una anotación. La diferencia más notable radica en que se arrastran anotaciones desde el documento HTML, ya que antes se arrastraban solamente los conceptos de las ontologías, debido al uso de una anotación como valor para un elemento (el destino de la relación).

Utilizando el panel inferior derecho, se deben rellenar los campos "Origin" (origen), "Relationship" (relación), y "Destination" (destino). El campo relación se rellena, de forma inmediata, al seleccionar un concepto de una ontología cuyo tipo sea de relaciones. Para rellenar los valores origen y destino, basta con seleccionar una anotación, como se vio en la sección "Seleccionar una anotación". Finalmente, se debe pulsar el botón *Add* del subpanel de relaciones para que la relación se cree efectivamente.

Utilizando el botón secundario del ratón sobre una anotación, aparecerá una lista desplegable donde elegir una de las anotaciones disponibles. Tras elegir la anotación aparecerá un menú contextual (*pop-up menu*) donde se debe escoger "Add Relation". Acto seguido, se creará una

nueva fila en la tabla de relaciones lingüísticas. Para introducir la relación, se arrastrará desde una ontología de relaciones, hasta la fila previamente seleccionada y la casilla de la unidad destino se ocupa mediante el arrastre también, pero la forma de arrastrar una anotación a estas casillas es distinta a como se realiza con los conceptos de las ontologías. Arrastrar una anotación tiene distintos requisitos, ya que no se necesita que la fila a la que se va a arrastrar esté seleccionada, pero sí se requiere que se resalte la anotación antes de su arrastre a la tabla de relaciones. Para resaltar la anotación y que no se produzca una nueva anotación sobre esta, se debe "deseleccionar" el concepto activo. Esto último se consigue pulsando el botón *Unselect* de la pestaña "Selection", que se encuentra debajo del panel con las ontologías. Para asegurarse de que no existe ningún concepto activo, se puede observar el campo que se encuentra a la derecha del botón *Unselect*, donde se mostrará el nombre del concepto activo. Si no hay ninguno activo, el campo de texto aparecerá vacio. Cuando se haya resaltado el texto de la anotación que se quiere completar, y se suelte sobre la casilla deseada aparecerá un menú que puede dar a escoger entre, quizás, varias anotaciones. Este menú tiene su razón de existir, ya que puede haber dos o más anotaciones solapadas, es decir, que estén asociadas a mismo texto en idéntica ubicación. Cuando todos los campos de una fila estén completos, finalizará el proceso, y la relación será almacenada en el modelo.

Figura 14: Botón *Unselect* y campo que muestra el concepto activo.



El último modo para agregar una relación entre unidades anotadas, requiere tener activa la pestaña de relaciones lingüísticas. Realizando doble clic sobre una anotación, se desplegará una lista donde escoger la anotación que será el origen de la relación. Entonces aparecerá una nueva fila en la tabla de relaciones y habrá que completarla como se ha visto en el modo del párrafo superior.

ELIMINAR UNA ANOTACIÓN

Algunas veces el usuario puede querer eliminar una anotación, ya sea por un fallo en el manejo del ratón, al haber seleccionado un texto distinto al que se quería, o simplemente porque más tarde se da cuenta de que no es necesario o es incorrecta. Para eliminar una anotación se puede hacer clic sobre ella con el botón secundario del ratón. Aparecerá entonces una lista desplegable, donde se mostrarán todas las anotaciones que se encuentran en esa zona. Después de seleccionar una, aparecerá un menú contextual *(pop-up*

menú) en el que se debe elegir la opción "Remove Annotation". También se puede eliminar la anotación actualmente seleccionada, la que aparece en el campo "Unit", a través del elemento "Remove annotation" del menú "Annotation". Cuando se elimina una anotación, también se eliminan todas las filas de las tablas (tanto la de unidades, como de la de relaciones) que tienen alguna referencia a esta anotación que se desecha.

Existe la opción de eliminar todas las anotaciones existentes en el documento actual. Para ello, se debe hacer clic sobre el elemento "Remove all" del menú "Annotation". De esta forma, el documento quedará en el mismo estado que se encontraba antes de iniciar el proceso de anotado.

ELIMINAR FILAS DE LAS TABLAS

Igualmente, se puede desear eliminar un par atributo-valor de una anotación o una relación entre anotaciones, en vez de una anotación con todas sus dependencias. Tanto los pares atributo-valor como las relaciones entre anotaciones son mostrados en las tablas. Para eliminar una fila de la tabla de unidades lingüísticas, se debe seleccionar la misma y pulsar el botón *Delete attribute-value*; para eliminarla de la tabla de relaciones lingüísticas, también se debe escoger la fila que representa la relación que se va a descartar y pulsar el botón *Delete relation*.

CARGAR UNA ONTOLOGÍA UNA VEZ INICIADA LA SESIÓN

Puede suceder que se quiera añadir una nueva ontología mientras se está anotando, a pesar de que antes de iniciar la sesión se pudieron indicar las ontologías escogidas para anotar. Lo primero que hay que hacer para cargar una ontología, una vez iniciado el proceso de anotado, es seleccionar "Load ontology" en el menú "File". Lo segundo es buscar y cargar la ontología mediante un selector de ficheros. En tercer lugar, habrá que indicar un

nombre para la ontología. Y como cuarto y último paso, se dotará de un tipo a la ontología.

CAMBIAR DE DOCUMENTO Y GUARDAR LAS ANOTACIONES

Para finalizar el manual de usuario, se mostrará cómo cambiar de documento para anotar otro, y cómo almacenar los datos en un fichero de salida con formato OWL.

Inicialmente, se indicó un corpus, que constaba de los diferentes documentos que se han de anotar, pero de momento sólo se ha anotado un mismo documento. Para cambiar de documento, siempre dentro del corpus especificado al inicio, se dispone de unos botones en la barra de herramientas. Estos botones son los cuatro que se encuentran más a su izquierda, y cuyos iconos son flechas. Según el orden, de izquierda a derecha, tienen las siguientes funciones:

1. Este botón establece el documento primero como actual y lo muestra.
2. Este botón establece el documento previo como actual y lo muestra.
3. Este botón establece el documento siguiente como actual y lo muestra.
4. Este botón establece el último documento como actual y lo muestra.

Cuando se quiera guardar las anotaciones realizadas, se deberá hacer documento por documento, es decir, se deciden los documentos para los que se quieren conservar las anotaciones una vez cerrada la aplicación. Almacenar las anotaciones de un documento determinado requiere que este sea el documento actualmente mostrado. De esta forma, lo primero que debe realizar el usuario es, cambiar al documento del que se quieren conservar las anotaciones. Una vez que se tiene el documento deseado como actual, basta con pulsar el botón de la barra de herramientas que muestra un disquete. Al pulsar el botón, aparecerá una ventana con una barra de progreso. Mientras dure el proceso de guardado se mostrará esta ventana, aunque se puede seguir realizando cambios, ya que es un

proceso paralelo. Sin embargo, los cambios realizados mientras se guardan las anotaciones no se reflejarán en el fichero de salida.