Asistente para el depósito de documentos en Repositorios utilizando extracción semiautomática de metadatos

Tesina de Grado

Licenciatura en Ciencias de la Computación

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Universidad Nacional de Rosario

Autor: Santiago Fontanarrosa
Directoras: Dra. Ana Casali
Dra. Claudia Deco

10 de septiembre de 2015

Resumen

En este trabajo se propone facilitar el proceso de depósito de objetos digitales educativos en repositorios modificando el flujo de carga estándar de plataformas tales como DSpace. Además, se ha propuesto una arquitectura de un Asistente para la Extracción Automática de algunos metadatos de los documentos. Estos metadatos extraídos automáticamente son validados por el usuario en el proceso de descripción del objeto. Para el asistente, se analizaron distintas herramientas de extracción y en particular se propuso utilizar una combinación de las mismas. Un prototipo de este asistente se implementará en el Repositorio RepHip de la Universidad Nacional de Rosario. De esta forma se espera ayudar al usuario en el proceso de carga disminuyendo así su trabajo y mejorando la cantidad y la calidad de los metadatos cargados.

Agradecimientos

"Un viaje de mil millas comienza con el primer paso"
— Lao-tsé

Este trabajo esta dedicado a todos los que me acompañaron a lo largo de este viaje que hoy llega a su fin. Que me ayudaron a seguir adelante, cuando parecía que me rendía; que creyeron en mi, más de lo que yo lo hacia en mi mismo.

Que me dieron la confianza y la seguridad de que al final del camino, se encuentra la satisfacción de haber logrado algo importante.

Índice general

Índice de figuras

Capítulo 1

Introducción

El desarrollo de Repositorios Institucionales de Acceso Abierto es prioritario para la preservación y diseminación de conocimiento abierto y accesible para toda la ciudadanía. En particular, la creación de repositorios de objetos digitales educativos en las universidades públicas de Argentina, es una prioridad en el marco de las políticas del Ministerio de Ciencia, Tecnología e Innovación y el Consejo Interuniversitario Nacional. El objetivo de estos repositorios es viabilizar de una forma eficiente el almacenamiento, clasificación, búsqueda y reutilización de estos recursos educacionales.

Con el fin de diseñar un repositorio institucional, se ha observado que la comunidad de docentes-investigadores de la región propone en sus prácticas académicas diferentes tipos de producción y simultaneidad de campos de aplicación de un mismo objeto digital. En este sentido, se propone un repositorio universitario integrador de "Objetos Digitales Educativos" donde una publicación científica o una obra de arte pueden ser también consideradas objetos de aprendizaje [San Martín et al., 2013].

En particular la UNR ha creado en los últimos años un Repositorio Hipermedial institucional, denominado RepHip [1] cuyo objetivo es almacenar toda la producción académica, científica y de extensión de la UNR. Este repositorio está implementado en la plataforma DSpace [2]. A partir de políticas nacionales elaboradas en esta dirección, se ha aprobado y ejecutado el Proyecto de la Agencia Nacional de Promoción Científica y Tecnológica, Convocatoria PICTO CIN II "Hacia el Desarrollo y Utilización de Repositorios de Acceso Abierto para ODE en el Contexto de las Universidades Públicas de la Región Centro-Este de Argentina. (Resolución ANPCYT Nº 330/2011, ejecución 2012-2013), dentro del cual se en-

[1] http://rephip.unr.edu.ar
[2] dspace.org

marca esta Tesina.

Para ayudar al almacenamiento, clasificación, búsqueda y reutilización de los recursos educacionales, surge el concepto de Objetos de Aprendizaje (Learning Objects - LO). Éstos pueden ser usados por un estudiante que quiere aprender un determinado tema o por un profesor que quiere preparar algún material para su clase. Los usuarios pueden recuperar estos objetos por medio de búsquedas en repositorios Web. Los Repositorios Institucionales almacenan la producción docente, científica y de extensión, y permiten una búsqueda más acotada para la recuperación y reutilización de estos recursos digitales.

Estos objetos se almacenan utilizando metadatos descriptivos que proporcionan información adicional sobre los mismos. La información almacenada en estos metadatos es fundamental para la mejor recuperación de los mismos y se vuelven un aspecto clave en el rendimiento y calidad de los objetos retornados. Existen distintos estándares de metadatos tales como DublinCore [3] y IEEE LOM [4], que utilizan distintas categorías no sólo para describir el contenido del objeto (título, autor, palabras claves, idioma, etc..) sino también, como en el caso de LOM, permiten describir aspectos educacionales de los mismos (nivel educativo, complejidad, etc.). Sin embargo, en la mayoría de los casos, la calidad de la información cargada en estos metadatos en los distintos repositorios, es de baja calidad o incompleta [Sonntag, 2004][Casali et al., 2011]. Esto se debe a que la carga de metadatos, es una tarea que suele ser tediosa, que consume tiempo y muchas veces las personas encargadas de llenar los mismos, deciden no hacerlo.

Para facilitar la carga de objetos digitales educativos en el repositorio se propone modificar el flujo de carga estándar de la plataforma DSpace y diseñar un asistente de carga para la extracción automática de algunos metadatos. De esta forma se ayudará al usuario en este proceso disminuyendo su trabajo y mejorando la cantidad y la calidad de los metadatos cargados. En este trabajo se presenta el nuevo flujo para el depósito de objetos, se propone una arquitectura de este Asistente y el diseño de un prototipo para la carga de objetos en repositorios desarrollados sobre DSpace.

[3] http://dublincore.org
[4] http://www.ieee.org

Capítulo 2

Conceptos Preliminares

2.1. Objetos de Aprendizaje y Repositorios

En la actualidad, no existe un consenso dentro de la comunidad científica respecto a qué es un "Objetos de Aprendizaje". De acuerdo con Wiley [Wiley, 2003]:

> "Un objeto de aprendizaje es cualquier recurso digital que puede ser reutilizado para la enseñanza"

Por su lado, el "Institute of Electrical and Electronics Engineers (IEEE)"[1] lo define como [Duval, 2002]:

> "Entidad, digital o no digital, que puede ser usada para aprendizaje, educación o entrenamiento"

En el contexto del presente trabajo, los objetos de aprendizaje son todos aquellos materiales digitales que como unidad o agrupación permiten y/o facilitan un objetivo educacional. Estos objetos luego pueden ser usados tanto por un estudiante que quiere aprender un determinado tema o por un profesor que quiere preparar algún material para su clase.

Los usuarios pueden recuperar estos objetos por medio de búsquedas en repositorios Web. Los Repositorios Institucionales almacenan la producción docente, científica y de extensión, y permiten una búsqueda más acotada para la recuperación y reutilización de estos recursos digitales. Algunos ejemplos de estos repositorios son:

1. LA FLOR (Latin American Federation of Learning Object Repositories): `http://laflor.laclo.org`

[1] http://www.ieee.org

2. Ariadne: `http://www.ariadne-eu.org`

3. OER Commons: `https://www.oercommons.org`

4. MERLOT: `http://www.merlot.org`

2.2. La plataforma DSpace

La plataforma DSpace®, surge a partir de la colaboración entre la compañía Hewlett-Packard™y la Biblioteca del Instituto Tecnológico de Massachusetts [2] (Massachusetts Institute of Technology (MIT)), con el fin de resolver la siguiente problemática:

> [...] A medida que profesores y otros investigadores han comenzado a desarrollar materiales de investigación y publicaciones académicas en formatos digitales cada vez más complejos, surge la necesidad de reunirlos, conservarlos, indexarlos y distribuirlos: una tarea que consume mucho tiempo y resulta costosa de gestionar de forma individual por cada uno de ellos, así como también para los departamentos, laboratorios y centros de investigación a los cuales pertenecen.

El sistema DSpace proporciona una manera de manejar estos materiales de investigación y publicaciones en un repositorio que logra el mantenerlos de forma profesional y permite darles mayor visibilidad y accesibilidad en el tiempo [Smith et al., 2003]. En la Figura 2.1 se muestra la arquitectura de la plataforma.

2.3. Metadatos

Estos objetos se almacenan utilizando metadatos descriptivos que proporcionan información adicional sobre el mismo. La información almacenada en estos metadatos es fundamental para la mejor recuperación de los mismos y se vuelven un aspecto clave en el rendimiento y calidad de los objetos retornados. Existen distintos estándares de metadatos tales como DublinCore[3] y IEEE LOM[4], que utilizan distintas categorías no sólo para describir el contenido del objeto (título, autor, palabras claves, idioma, etc..) sino también, como en el caso de LOM, permiten describir aspectos educacionales de los mismos (nivel educativo, complejidad, etc.). Sin embargo, en la mayoría de los casos, la calidad de la información cargada en estos metadatos en los distintos repositorios, es de baja

[2] http://libraries.mit.edu
[3] `http://dublincore.org`
[4] `http://www.ieee.org`

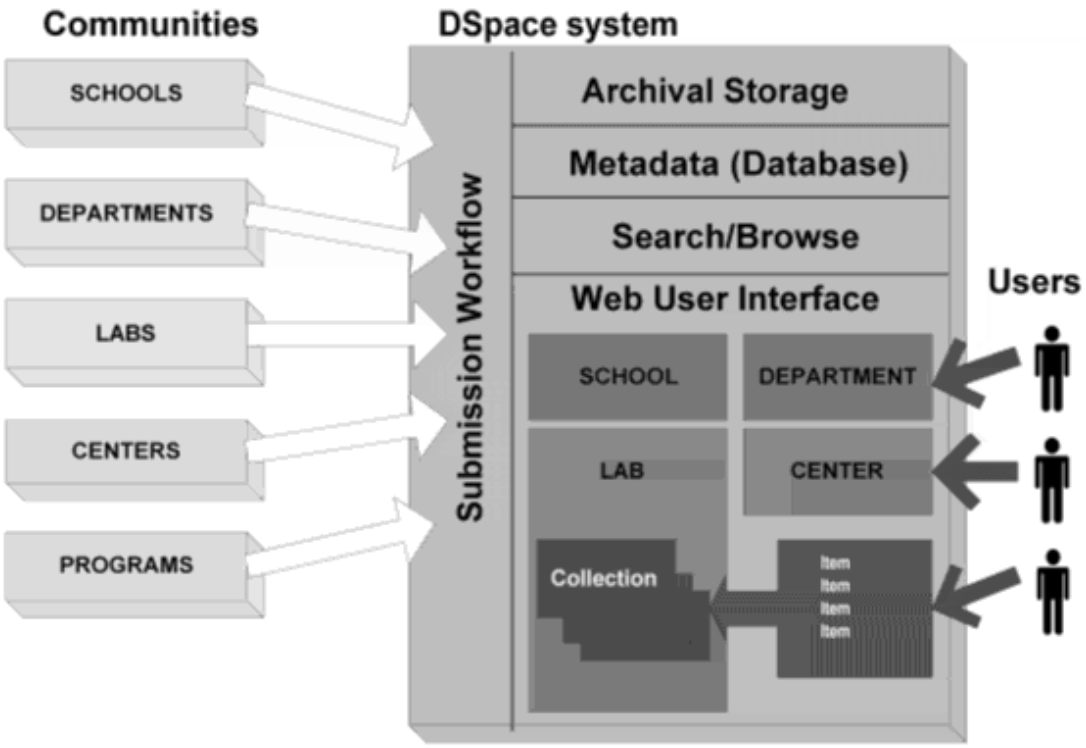

Figura 2.1 – *Arquitectura DSpace*

calidad o incompleta [Sonntag, 2004][Casali et al., 2011]. Esto se debe a que la carga de metadatos, es una tarea que suele ser tediosa, que consume tiempo y muchas veces las personas encargadas de llenar los mismos, deciden no hacerlo.

DSpace utiliza un estándar de metadatos Dublin Core cualificado para la descripción de documentos de bibliotecas (más concretramente el "Libraries Working Group Application Profile"[5]). Sólo tres campos son requeridos: título, idioma y fecha de entrega, todos los otros campos son opcionales. Hay campos adicionales para documentos resúmenes, palabras clave, metadatos técnicos y metadatos de los derechos, entre otros.

Estos metadatos se muestran en el registro de artículo en DSpace, y están indexado para navegar y buscar en el sistema (dentro de una misma colección, entre distintas colecciones, o entre distintas Comunidades). Para los Paquetes de Difusión de Información (DIPs)[6] los cuales forman parte del framework OAIS[7], el sistema actualmente exporta metadatos y material digital en un esquema XML.

[5] `http://dublincore.org/documents/library-application-profile`
[6] `http://www.iasa-web.org/tc04/formats-and-dissemination-information-packages-dip`
[7] `https://www.oasis-open.org/`

2.4. Métricas de Evaluación

Es importante el poder contar con métricas que permitan medir que "tan bueno" es el extractor en relación con la información que se dispone en el documento.

Las dos métricas que se utilizaron en el siguiente trabajo se encuentran descriptas en [Ting, 2010]:

Precisión: puede ser entendida como la fracción de datos correctos dentro de todos los recuperados

$$P = \frac{\#Respuestas\ Correctas}{\#Respuestas\ Producidas} \tag{2.1}$$

Cobertura: puede ser entendida como la fracción de datos correctamente extraídos con respecto a todos los datos disponibles

$$C = \frac{\#Respuestas\ Correctas}{\#Total\ Posible\ de\ Respuestas\ Correctas} \tag{2.2}$$

Ambas toman valores siempre dentro del intervalo [0,1], siendo su óptimo 1. Los casos bordes suceden cuando se tiene un denominador nulo, tanto para P como para C. Para C, cuando $\#Total\ Posible\ de\ Respuestas\ Correctas = 0$, el cómputo resulta indeterminado (NaN). La convención usual al respecto es tomar C = 1 en caso de que $\#Respuestas\ Producidas = 0$ y $C = 0$ en caso contrario.

Lo que se busca es poder "premiar" o "penalizar" el hecho de haber producido o no datos espurios ante la ausencia de datos. Por otra parte, cuando $\#Respuestas\ Producidas = 0$ se asigna un valor no numérico a P, considerando que carece de importancia medir la precisión en esta situación.

Capítulo 3

Flujo de Carga

3.1. Arquitectura del Flujo de Carga en DSpace

DSpace utiliza Colecciones para agrupar los documentos, las cuales son manejadas por un Administrador de la Comunidad/Subcomunidad. Para cargar un documento, el usuario selecciona una comunidad, y en base a esa selección se pueden determinar los tipos de objeto digitales que el usuario podrá elegir. Por ejemplo, la comunidad Departamento de Ciencias de la Computación tiene asociadas las colecciones Tesinas, Artículos, Comunicaciones a Congresos y Materiales Educativos.

El proceso de carga por defecto de objetos en repositorios gestionados por DSpace se divide en una serie de pasos. En estos pasos se eligen las colecciones en las que se va a realizar el depósito, se describe el objeto mediante metadatos, se suben los archivos que lo componen, se acepta la licencia del repositorio y se hace una revisión previa a que se complete el depósito. En la Figura 3.1 se muestra el esquema de dicho proceso.

También, se puede agregar un paso extra que permite elegir una licencia Creative Commons para el objeto a depositar antes de la revisión. Cada una de estas tareas debe realizarse en el orden mencionado. La descripción del objeto mediante metadatos se realiza en tres etapas: una de preguntas iniciales y otras dos para cargar los metadatos obligatorios y los metadatos opcionales del objeto.

A partir de un análisis de usabilidad de DSpace se encontraron los siguientes problemas en el proceso de depósito:

- **La tarea de carga de metadatos es tediosa:** la descripción con metadatos del objeto a depositar es un gran cuello de botella en el depósito ya

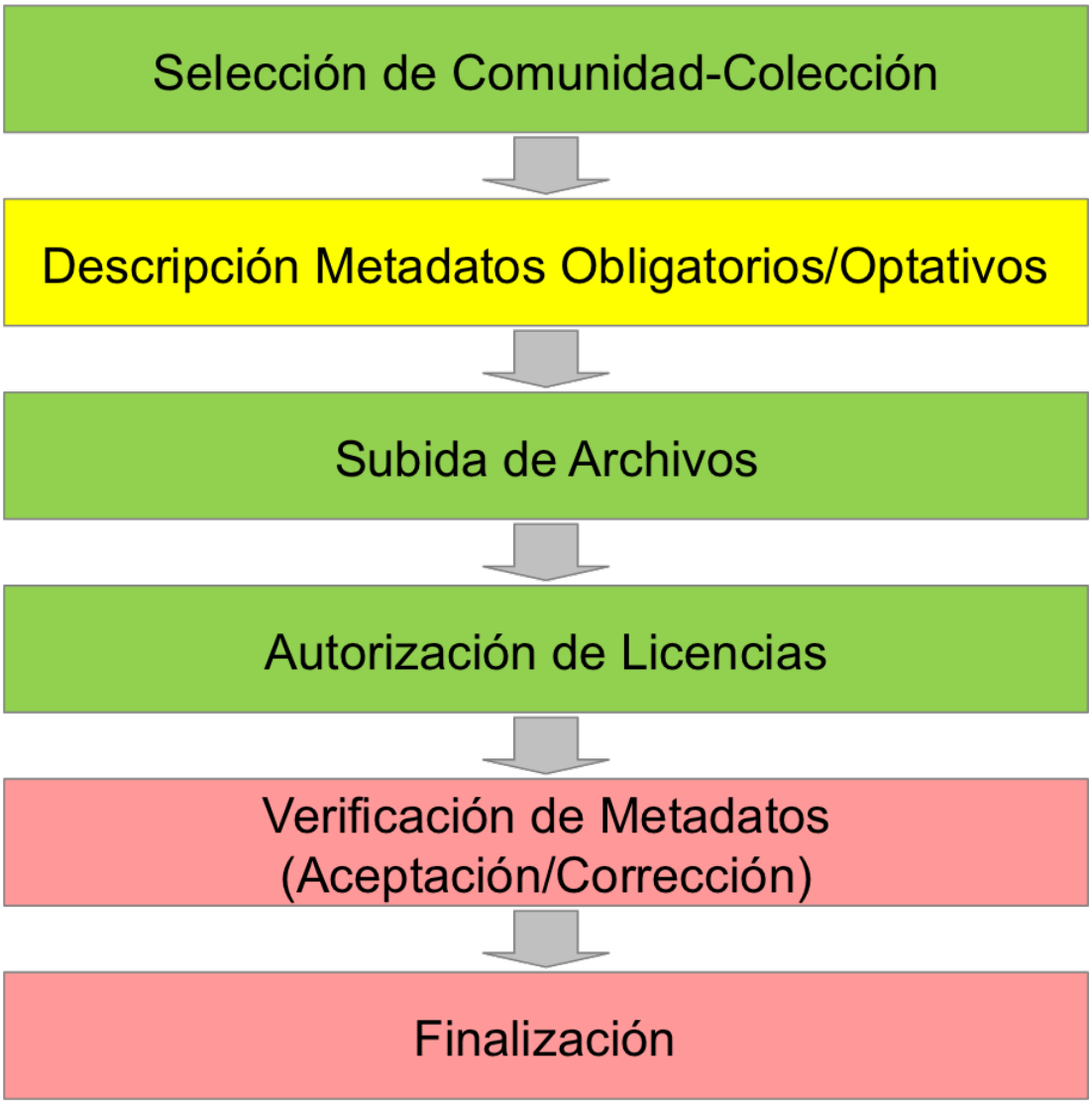

Figura 3.1 – *Proceso de carga*

que mucha de la información requerida no está al alcance inmediato de la persona que realiza el depósito y no se completa.

- **La interfaz del flujo de depósito no es clara:** existen problemas de interfaz que hacen confuso el depósito.
- **Algunos de los problemas encontrados son:** no es claro cómo comenzar una nueva carga, a pesar de ser una de las funcionalidades principales del repositorio; la elección de colecciones en donde éste se va a realizar es confusa ya que no especifica a qué comunidad pertenecen; los metadatos obligatorios no están diferenciados de forma clara de los metadatos opcionales; los distintos pasos de descripción tienen el mismo nombre y algunos botones del flujo de carga son ambiguos.

- **Los pasos para la descripción del objeto no son customizables según los distintos tipos de colecciones:** si bien DSpace permite la customización de los formularios de descripción a partir de un archivo de configuración simple, esta customización debe hacerla un administrador para cada colección; además, no permite agrupar colecciones en distintos tipos y elaborar depósitos customizados para cada tipo.

A continuación se detallan las mejoras para resolver las limitaciones expuestas.

3.2. Mejoras Planteadas

Para solucionar problemas anteriormente planteados, en este trabajo se propone reestructurar el flujo de carga, reordenando y modificando los pasos de depósito, modificar la interfaz e incorporar un asistente de extracción de metadatos de los archivos depositados. Las modificaciones principales al flujo de carga consisten en:

1. Elección de la colección: el nuevo paso muestra la estructura completa de comunidades y colecciones en las que se tiene permisos de realizar el depósito.

2. Aceptación de la licencia institucional: este paso se agrega al principio ya que en el caso de que ésta no se acepte, el depósito se cancela y los otros pasos no son necesarios.

3. Carga de los archivos asociados al objeto: se realiza antes de completar los formularios de descripción, de manera que algunos metadatos puedan completarse automáticamente con ayuda del asistente de extracción de metadatos.

4. Reordenamiento de la descripción mediante dos pasos bien diferenciados: uno para la carga de metadatos obligatorios y otro para metadatos opcionales. Estos metadatos cambian según el tipo de objeto que se está depositando y es inferido por la colección que se eligió. Los metadatos que fueron obtenidos por el asistente de extracción son mostrados para su validación por parte del usuario, en los campos correspondientes del formulario.

En la Figura 3.2 se muestra un esquema del flujo de carga con las mejoras que se plantean. En la Sección 5.1 se analiza en detalle las modificaciones realizadas al flujo de carga con el fin de integrar con el asistente de carga planteado.

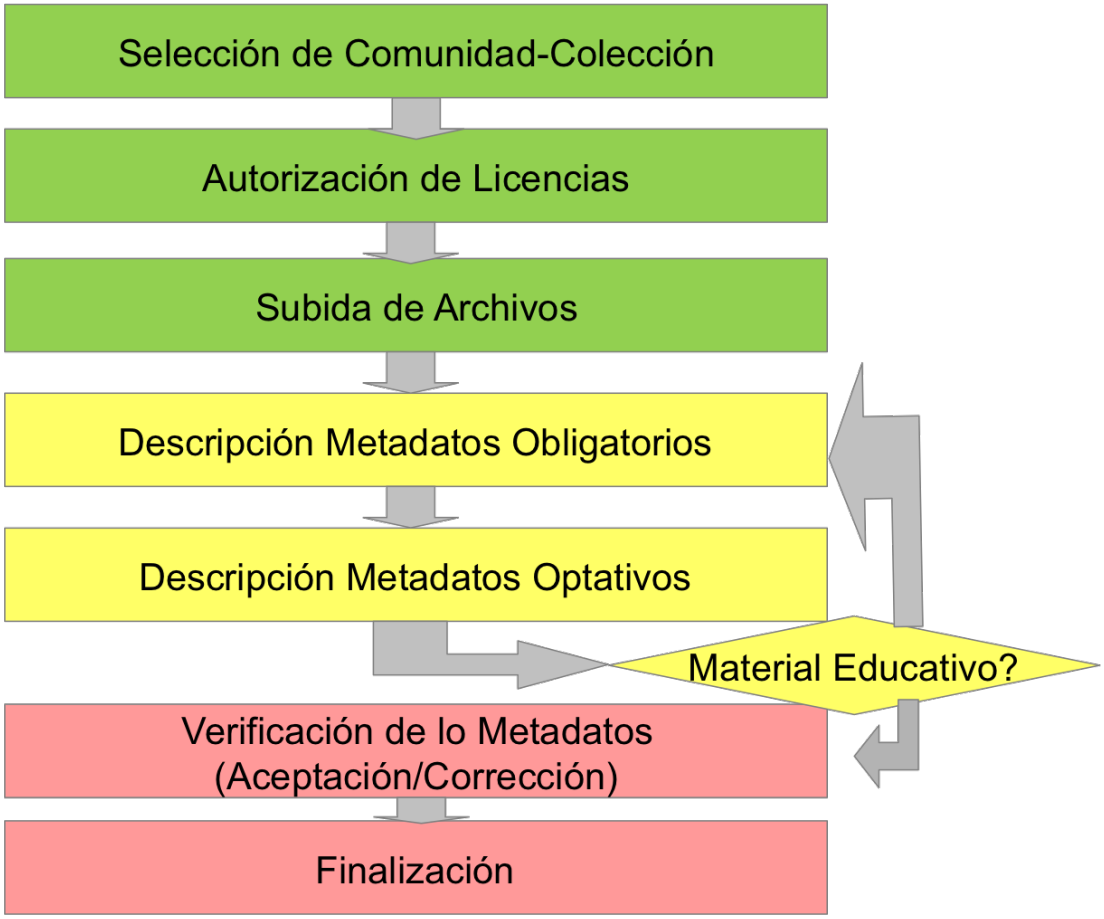

Figura 3.2 – *Proceso de carga*

Capítulo 4

Extracción de Metadatos

4.1. Análisis de Herramientas

A fin de poder incorporar la extracción automática o semiautomática de metadatos en el proceso de carga de documentos en el repositorio, se analizó qué herramientas extractoras o combinación de ellas se podría utilizar para poder obtener mejores resultados. No hay muchos trabajos focalizados en la extracción automática de metadatos, cada herramienta extrae distintos tipos de metadatos, tiene sus propios objetivos, arquitectura y usa distintas técnicas. En [Pire et al., 2011] se analizan cuatro sistemas dedicados a la extracción automática de metadatos educativos de objetos de aprendizaje: SAXEF [Alfano et al., 2007], TWYS [Yuen, 2007], Looking4LO [Motz et al., 2009] y MAGIC [Li et al., 2005]. Estas propuestas son relevantes, aunque algunas no están implementadas o no están disponibles como herramientas libres. Luego, se consideraron otras herramientas extractoras de metadatos generales tales como el título, los autores, las palabras claves, el resumen y el idioma. Para un primer análisis se seleccionaron las siguientes herramientas:

- AlchemyAPI
- KEA Automatic Keyphrase Extraction
- Mr Dlib
- ParsCit

4.2. AlchemyAPI

AlchemyAPI[1] es una plataforma de minería de texto la cual proporciona un conjunto de herramientas que permiten el análisis semántico utilizando técnicas de

[1] `http://www.alchemyapi.com`

procesamiento de lenguaje natural y machine learning, más precisamente algoritmos de "deep learning"[Deng and Yu, 2014]. Provee un conjunto de servicios que permiten analizar de forma automática documentos de texto. La herramienta expone varios servicios a partir de su RESTful API (http://www.alchemyapi.com/api/calling.html), entre los que se encuentran:

- identificación del autor
- identificación de entidades
- generación de palabras claves
- categorización del contenido
- identificación del idioma

En su versión gratuita, el servicio presenta una limitación de 1000 consultas diarias y un límite por consulta de 150 kbs.

4.3. KEA Automatic Keyphrase Extraction

KEA[2], es la implementación en JAVA del algoritmo KEA [Witten et al., 1999]. La herramienta extrae automáticamente frases claves del texto completo a partir del documento a analizar. El conjunto de todas las frases seleccionadas en un documento se identifican utilizando procesamiento léxico rudimentario. Utiliza técnicas de machine-learning para generar un clasificador que determina qué frases candidatas deben ser asignadas como frases clave. Esta herramienta puede ser utilizada en forma local y se necesita una fase previa de entrenamiento.

4.4. Mr Dlib

Mr Dlib[3] es una biblioteca digital que proporciona acceso a varios millones de artículos de texto completo y sus metadatos en formato XML y JSON a través de un servicio web RESTful. En su etapa beta de desarrollo, sus funcionalidades son utilizadas por terceros y permite extraer Título y Autores [Beel et al., 2011]

4.5. ParsCit

ParsCit [4] es una aplicación de código abierto que realiza dos tareas: el análisis sintáctico de cadenas de referencia, también llamado análisis de citas o extracción

[2] `http://www.nzdl.org/Kea/index_old.html`
[3] `http://www.mr-dlib.org`
[4] `http://wing.comp.nus.edu.sg/parsCit/`

de citas, y el análisis de la estructura lógica de documentos científicos. Estas tareas las realiza a partir de un archivo de texto plano utilizando procedimientos de aprendizaje automático supervisado que usan campos aleatorios condicionales (CRF) como mecanismo de aprendizaje. Incluye utilidades para ejecutarse como un servicio Web o como una aplicación independiente [Councill et al., 2008].

4.6. Selección del Extractor

A fin de determinar que componentes utilizar para el asistente de carga, se realizaron distintas pruebas con las herramientas mencionadas, sobre un corpus de 760 documentos del repositorio RepHip seleccionados según los siguientes criterios:

- diversidad temática
- colecciones a las que pertenece el archivo
- formato del archivo. E.j: PDF,PPT,texto plano,etc

Las pruebas que se realizaron buscaban evaluar los resultados obtenidos al realizar la extracción de títulos, autores, palabras claves e idioma así como el tiempo de respuesta, ya que es una condición que los resultados se obtengan en tiempo real. Se analizaron los resultados obtenidos por Kea, Alchemy y Mr. DLib, respecto a los distintos metadatos que pueden ser extraídos por cada uno ellos: Mr. DLib para título y autores, KEA para palabras claves y Alchemy para la extracción de Título, Palabras Claves e Idioma (Tabla 4.1):

	Titulo	Autor	Palabras Claves	Idioma
Kea			x	
Mr Lib	x	x		
AlchemyAPI	x		x	x

Tabla 4.1 – *Componentes evaluados por herramienta*

Mr Dlib : se realizaron varias pruebas: enviando las primeras, 1, 2 ó 4 páginas del archivo PDF al servidor de MrDlib para la extracción.

- El objetivo fue analizar si se obtienen extracciones más precisas y si los tiempos de procesamiento eran aceptables.

- Se obtuvo una precisión 65 % para los títulos y 40 % para autores
- Los resultados son similares, salvo que los tiempos de procesamiento por artículo en el caso de 4 páginas se duplican (los promedios oscilan entre los 5,6 y 11,2 seg).

KEA : se lo configuró para sugerir 5 palabras clave por documento y se compararon las "Palabras Clave extraídas"vs "Palabras Clave previamente ingresadas"

- El promedio de coincidencias fue de un 60 % con un tiempo de extracción promedio por documento de 1,79 segundos, basado en el corpus de documentos con tamaños de hasta 800 KBytes.

AlchemyAPI : las pruebas consistieron en submitir los archivos al servidor que provee AlchemyAPI, a fin de obtener la siguiente metadata: Tópico principal, Palabras Claves e Idioma.

- En la extracción del idioma del documento se obtuvo un 76 % de asignaciones correctas.
- Respecto a las palabras claves, Alchemy retorna un ranking de relevancia. Para su evaluación, se consideraron únicamente las primeras 5, y se compararon los resultados obtenidos con las cargadas por el autor del documento. Se obtuvo un 56 % de resultados correctos, donde se recuperaron todas o algunas de las palabras claves cargadas por el autor.
- Considerando el tópico o tema principal del documento, se analizó la respuesta que devuelve el servicio, en la forma de una única categoría, y se lo comparó con el título del documento que estaba siendo analizado para evaluar si existía coincidencia. En este caso el 67 % de las asignaciones de categoría fueron correctas.

	Titulo	Autor	Palabras Claves	Idioma
Kea			60 %	
Mr Lib	65 %	40 %		
AlchemyAPI	67 %		56 %	76 %

Tabla 4.2 – *Precisión obtenida por cada herramienta*

Si bien los resultados de extracción de títulos y autores de los documentos, utilizando Mr. DLib fueron parcialmente satisfactorios (precisión 65 % para los títulos y 40 % para autores) se lo descartó para este primer prototipo de Asistente ya que esta herramienta estaba en una etapa muy temprana de desarrollo y su accesibilidad no estaba garantizada.

A su vez, se observa que los resultados obtenidos utilizando tanto KEA como Alchemy respecto a palabras claves son similares en precisión y que los resultados obtenidos con Mr. DLib y Alchemy para título y autores, también lo son. Dado que la herramienta Alchemy también permite obtener el idioma, se plantea utilizarla en el Asistente combinándola con otra herramienta para el preprocesamiento de documentos como ParsCit para mejorar los resultados. En la Tabla 4.2 se enumeran los resultados obtenidos para cada extractor.

En una segunda etapa de prueba, considerando el mismo conjunto de documentos, se agregó un paso más en el proceso de extracción de palabras claves, ejecutando primero la herramienta ParsCit y luego Alchemy. ParsCit permite dar estructura al documento y genera en un documento xml en el cual intenta identificar: Título, Autor, Resumen y Palabras claves. Esta información se concatena en un nuevo archivo, el cual se utiliza para subir al servidor de AlChemyAPI en lugar del archivo original. Se obtuvo en este caso, el 70 % de resultados correctos, considerando como correctos aquellos donde se recuperaron todas o algunas de las palabras claves cargadas por el autor.

	Titulo	Autor	Palabras Claves	Idioma
Kea			60 %	
Mr Lib	65 %	40 %		
AlchemyAPI+ParsCit	67 %		70 %	76 %

Tabla 4.3 – *AlchemyAPI+ParsCit*

A partir de esta combinación de herramientas, se logró incrementar sustancialmente la calidad de las palabras claves retornadas pasando de un 56 % de resultados correctos obtenidos con Alchemy a un 70 % resultante con ParCit+Alchemy. En la Tabla 4.3 se listan los resultados obtenidos.

Al analizar los resultados erróneos, se observó que esto se debió a que algunos documentos no reportaron datos y/o no pudieron ser analizados. La imposibilidad de dicho análisis se debió a alguno de los siguientes problemas:

1. El archivo presentaba un formato que no permitía extraer y transformar el contenido del mismo a texto plano (por ej. archivos ppt convertidos a pdf) lo cual hacía que la calidad de la extracción fuera baja y/o incluyera la mayoría de metadatos propios del formato, los cuales no aportan valor real al extractor.

2. La cantidad de texto plano extraído del documento original no era suficiente para que el servicio de Alchemy pudiera generar una respuesta.

3. El formato del documento original no era soportado por la herramienta que debe transformarlo a texto plano.

A partir de los resultados obtenidos, se decidió utilizar ParsCit para dar estructura al documento, ademas de permitir la extracción "titulo" "author", y a AlchemyAPI para la generación palabras claves. Los detalles de la implementación se se encuentran en la Sección 5.1.

Capítulo 5

Asistente de carga

5.1. Arquitectura

Como se describió en el Capítulo 3, el proceso de carga de un documento en DSpace está compuesto de pasos, cuyo orden puede ser modificado tanto en el orden, como en el número de los mismos. La arquitectura que se plantea, incluye el desarrollar un "paso"[1] adicional el cual será responsable de:

1. Invocar al módulo de generación de metadata.
2. Obtener el resultado, pasando como parámetro el contenido del archivo que se está submitiendo.
3. Parsear los resultados y presentarlos de forma que sean consumibles.

El módulo de generación de metadata, está compuesto de los siguientes submódulos, los cuales tienen una arquitectura interna de "pipe-line", facilitando así el agregar nuevos módulos al proceso.

1. **File to Text**: este módulo toma como entrada el archivo que cargó el usuario. DSpace almacena dicho archivo en su representación de bytestream. Este módulo es responsable de transformarlo a texto.
2. **ParsCit**: este módulo recibe como entrada el archivo en texto plano e intenta organizar y dar estructura al mismo. Para ellos se utiliza la herramienta ParsCit[2]. Entre los resultados que provee podemos contar con:

   a) Título

   b) Resumen

[1] https://wiki.duraspace.org/display/DSDOC4x/Functional+Overview

[2] `http://wing.comp.nus.edu.sg/parsCit/`

c) Palabras claves

Esta información es expuesta en formato XML.

3. **Alchemy API**: este módulo recibe como entrada el archivo en formato xml con la estructura, y utiliza dicha estructura para enviar al servicio de metadata Alchemy, solamente el texto con mayor probabilidad de tener información que resulte relevante. El servidor retorna la respuesta en formato JSON; el módulo es responsable de formatear la respuesta y transformar en un formato reconocible por el proceso de carga de DSpace.

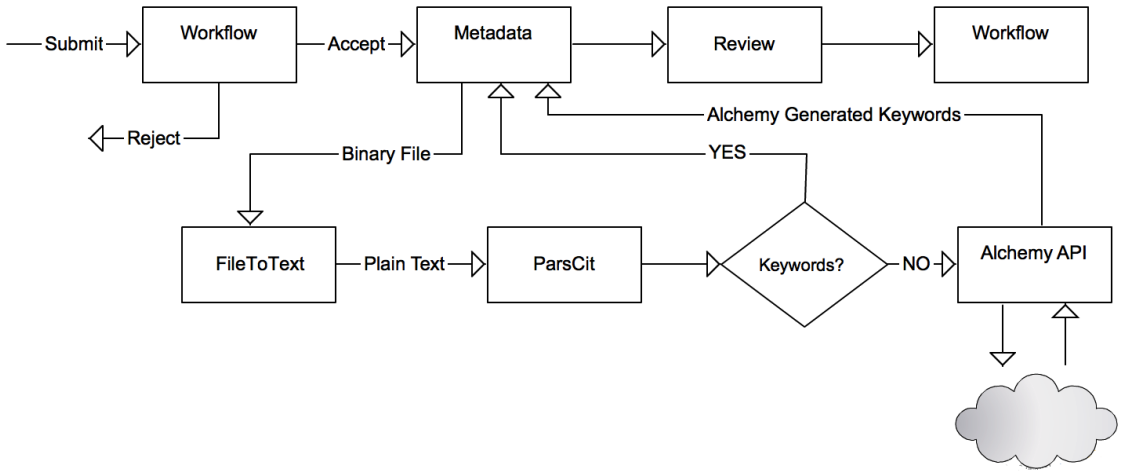

Figura 5.1 – *Arquitectura de Carga*

Existen casos en cuales el autor enumera de forma explicita en el documento el conjunto de palabras claves; un articulo puede incluir una sección "Palabras Claves" en el cuerpo del mismo. Como se puede observar en la Figura 5.1, cuando le es posible, el modulo **ParsCit** extrae y retorna como resultado dicho conjunto de palabras claves. En los casos en cuales no se pudo extraer un conjunto palabras claves valido, se procede a generarlo a partir del modulo **Alchemy API**.

Una vez finalizado el proceso de generación de metadata, el proceso de carga de documentos en DSpace, continua de forma normal, tal cual se describe en la Sección 3.1.

Nuevos módulos pueden ser agregados y/o modificados sin afectar el proceso de carga, permitiendo de esta forma el personalizar el proceso y adaptarlo a las diversas necesidades y/o especializar en base a el área de dominio del repositorio.

5.2. Desarrollo del Prototipo

Los 3 módulos que componen el generador de metadata, forman una unidad la cual tiene una única interfaz de entrada y una de salida. El "meta-modulo" que agrupa los distintos componentes, se encuentra desarrollado en Python.

El script toma como parámetros de entrada:

- un path a un archivo donde se encuentran el documento
- la colección a la que pertenece el archivo (definido en el contexto de DSpace)

```
$ python MetadataExtractor_DSpace.py -f ./ITEM@2133-453.txt -c LCC

  usage: MetadataExtractor_DSpace.py [-h] -f FILE -c COLLECTION

  -h, --help            show this help message and exit
  -f FILE, --file FILE  Path to file to be parsed
  -c COLLECTION, --collection COLLECTION
                        Collection name for the file
```

El formato de salida que genera el script se muestra a continuacion:

```
<?xml version="1.0" encoding="UTF-8"?>
<dspaceMetadata>
  <fileName>ITEM@2133-453.txt</fileName>
  <collection>Default</collection>
 <language>spanish</language>
 <category>arts_entertainment</category>
  <authors>
    <author confidence="0.929597">Sabina Florio</author>
  </authors>
  <titles>
    <title confidence="0.948077">Un lugar en la historia</
       title>
    <title confidence="0.557829">Modernity  Tradition
       Argentian art: Art from Rosario</title>
  </titles>
  <keywords>
    <keyword confidence="0.959272">Augusto Schiavoni</
       keyword>
    <keyword confidence="0.523035">showed which</keyword>
  </keywords>
</dspaceMetadata>
```

A fin de poder hacerlo compatible con DSpace, el nuevo "paso" del flujo de carga que generará el conjunto de metadatos (el cual fue desarrollado en el

lenguaje de programación Java) puede realizar una ejecución "externa", haciendo una llamada al sistema local con la llamada Runtime.getRuntime().exec

El script retorna la información generada en un archivo con formato JSON, el cual incluye, el título, idioma y el conjunto de palabras claves.

Capítulo 6

Experimentación

A fin de poder evaluar el prototipo implementado, se procedió a generar conjuntos de documentos tanto en idioma Ingles como en idioma Español, los cuales fueron extraídos del repositorio e-LIS[1].

Los documentos utilizados, fueron seleccionados de acuerdo al siguiente criterio:

- los documentos son referidos al área de la Bibliotecología
- son en formato PDF
- corresponden al formato de tesis y/o de artículo

Para cada uno de los documentos elegidos, se procedió a obtener el conjunto de palabras claves que los usuarios pre-cargaron, a fin de poder utilizarlos como casos de referencia para poder evaluar los resultados generados.

Dado que la carga y generación de metadata a través de la plataforma de DSpace requiere la intervención humana, y a fin de agilizar el proceso de evaluación, se procedió a automatizar las pruebas a través de la utilización de scripts que permiten realizar las tareas simulando la interacción del usuario. Las funcionalidades que se automatizaron se detallan a continuación:

Descarga de documento: a partir del listado de documentos obtenidos por los criterios de búsqueda en el repositorio, se procede a descargar cada uno de ellos. Los archivos son descargados a disco en su formato original (PDF) y se ingresa su registro correspondiente en la base de datos local de referencia. Adicionalmente, se le asocia la siguiente información: título del documento, autores y palabras claves que se cargaron por el usuario.

[1] `http://eprints.rclis.org`

Extracción del texto plano: una vez que se cuenta con el conjunto de documentos de prueba, se procede a extraer el texto en formato plano. Esto es necesario dado que el módulo de generación de metadatos (descripto en la Sección 5.1) acepta como parámetro de entrada el archivo en dicho formato. Para dicha tarea se utiliza la herramienta pdf2text[2].

Generación de los metadatos en esta instancia, se procede a invocar al módulo de generación de metadata, simulando que se estuviera haciendo dentro del proceso de carga normal de DSpace. El resultado es guardado a disco, para su posterior procesamiento

Procesamiento de resultados: todos los resultados son guardados de forma local utilizando el motor de base de datos MySQL[3]. Por cada archivo a evaluar, se registran todos los campos extraídos por la herramienta, para su posterior evaluación.

6.1. Análisis preliminar

Durante la primera etapa de experimentación, se procedió a ejecutar los casos de prueba sobre un conjunto reducido de documentos, a fin de evaluar la efectividad de las validaciones. Inicialmente se observó que ParsCit fallaba en detectar las palabras claves en los documentos, aún cuando éstas contaban con su propia sección y se especificaban de tal forma. Por ejemplo, con el término "Keywords" o "Palabras Claves" en un párrafo separado.

Analizando en mayor detalle los documentos en los cuales se observaba este comportamiento, se pudo detectar que dada la particularidad de algunos documentos, los cuales no seguían el formato tradicional de Tesis y/o Artículo, estas secciones eran identificadas erróneamente. En su mayoría esto se debía a que el formato que siguen algunas publicaciones no respetaban la misma segmentación y/o indentación, o en algunos casos el formato de múltiples columnas tendía a romper el "flujo" natural del texto, con lo cual el algoritmo fallaba en dar estructura.

Las mejoras realizadas consistieron en:

- Luego de la etapa de **"Extracción del texto plano"**, se procede a remover todas las ocurrencias de "stop words"[Wilbur and Sirotkin, 1992]. Esto

[2] `http://linux.die.net/man/1/pdftotext`
[3] `http://www.mysql.com`

permite reducir el número de falsos positivos en lo referido a la generación de las secciones correspondientes a palabras claves. Para dicha tarea se utilizó la funcionalidad que provee la librería "Natural Language Toolkit"(NLTK 3.0)[4]

- Una vez concluido este pre-procesamiento, el módulo de extracción, tomando como entrada el resultado que otorga Parscit, recorre las secciones generadas y intenta detectar ocurrencias de palabras que puedan hacer referencia a las "palabras claves"del documento, por ejemplo: "Keyword:","Palabras Claves:", etc.

A partir de estos cambios, se mejoró considerablemente la extracción de palabras claves en los documentos. Esta mejora, se notó especialmente en aquellos documentos que siguen una estructura "predecible", similar a la de una Tesis y/o Artículo. En la Figura 5.1 se puede observar el flujo modificado descripto.

6.2. Resultados primera fase de pruebas

En una primera instancia se generaron 2 conjuntos de documentos (Inglés y Español respectivamente) de 100 documentos cada uno, siguiendo los criterios mencionados al comienzo del capítulo y realizando el proceso de generación de datos descripto anteriormente.

Para cada uno de estos documentos se procedió a extraer los siguientes conjuntos de datos:

- Palabras Claves
- Autor/es
- Título
- Idioma

Luego se procedió a compararlos con los datos que los usuarios habían precargados en el repositorio; adicionalmente, se realizó una comparación "manual"de los resultados obtenidos cotejándolos con los archivos en cuestión. Para cada uno de los archivos se calculó la *Cobertura* y la *Precisión* por cada uno de los datos extraídos.

A continuación se muestran los resultados obtenidos para los 2 grupos.

[4] `http://www.nltk.org`

Conjunto Documentos en Inglés:

Se muestra a continuación el valor promedio de Cobertura y Precisión calculado sobre el conjunto de 100 documentos en Inglés. Los valores promedios están calculados para cada una de los elementos principales a extraer en los documentos: *Palabras Claves*, *Autor* y *Título*. Los mismos se muestran en la Tabla 6.1 y Figura 6.1.

	Cobertura	Precisión
Palabras Claves	0.72	0.73
Autor	0.48	0.51
Título	0.53	0.58

Tabla 6.1 – *Valores promedio de Cobertura y Precisión para documentos en Inglés*

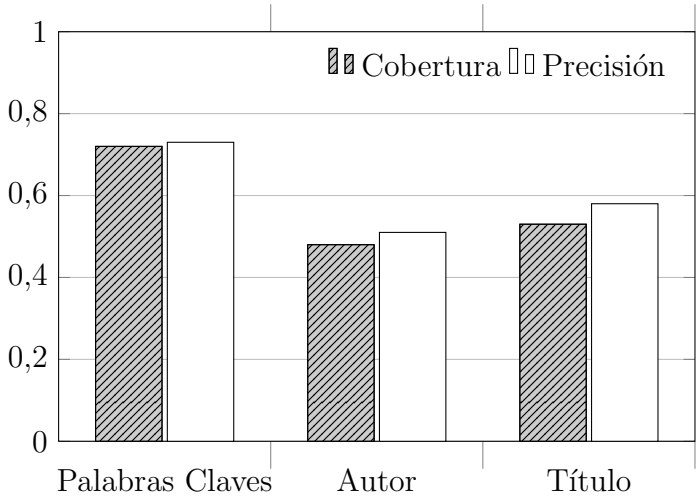

Figura 6.1 – *Valores promedio de Cobertura y Precisión para documentos en Inglés*

El detalle de los valores obtenidos, segmentados por intervalos, tanto para la Cobertura, como para la Precisión y agrupando por cantidad de archivos de cada segmento se detallan a continuación.

Palabras Claves:

En la Figura 6.2 se puede observar que hay 70 documentos cuya cobertura supera el 75 %, un resultado que se considera muy bueno.

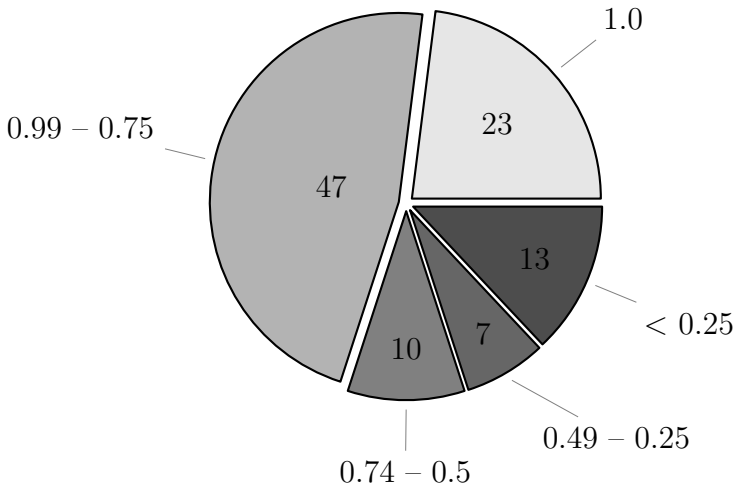

Figura 6.2 – *# de Archivos agrupados por rango de Cobertura obtenido para Palabras Clave*

Por otra parte, considerando la precisión, en la Figura 6.3 podemos observar que para 81 de los documentos se tienen valores superiores al 75 %, con lo cual se están obteniendo resultados que presentan información relevante al usuario final.

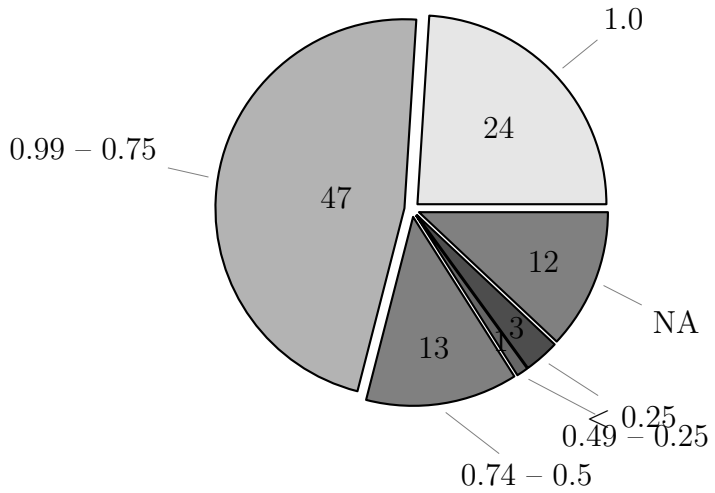

Figura 6.3 – *# de Archivos agrupados por rango de Precisión obtenido para Palabras Clave*

Como se describió en la Sección 2.4, cuando el numeró de respuestas producidas es 0, se considera que no tiene sentido medir la Precisión para dichos documentos. Dichos casos se marcan en la Figura 6.5 con la etiqueta NA (No Aplica).

Autor:

Con respecto a los resultados obtenidos para la extracción de Autores, tanto para la Cobertura, como para la Precisión, se obtuvieron muy buenos resultados para más de la mitad de los documentos, tal cual se puede observar en la Figura 6.4 y la Figura 6.5.

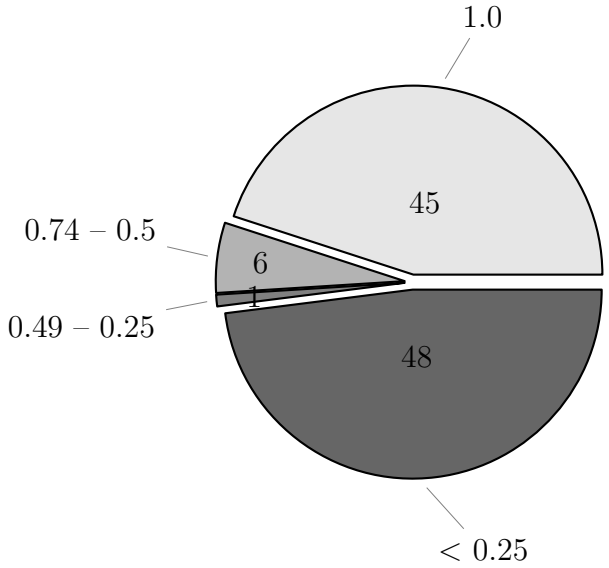

Figura 6.4 – *# de Archivos agrupados por rango de Cobertura obtenido para Autor*

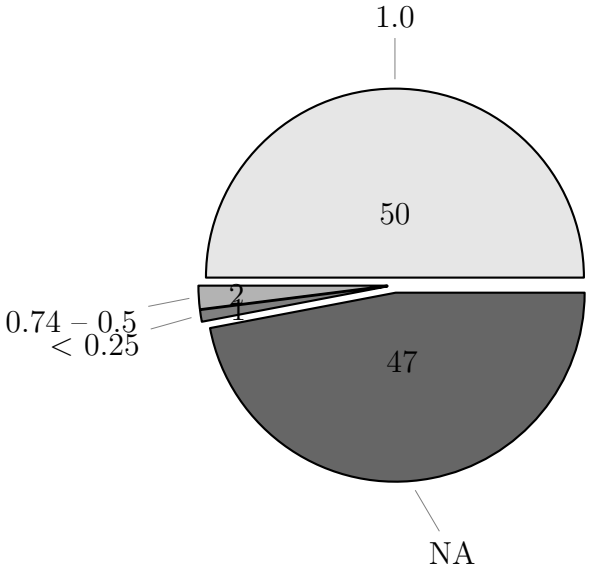

Figura 6.5 – *# de Archivos agrupados por rango de Precisión obtenido para Autor*

Título:

Con respecto a la extracción del Título, se partió de la presunción de que existe un solo resultado posible correcto, dado que los documentos por lo general solo tienen un título principal. Distinto es el caso del conjunto de palabras claves y/o autores, donde puede haber más de uno, y eso permite obtener resultados parciales.

Esto explica el rango tan acotado de valores en las métricas obtenidas cuando se procedió a medir la Precisión, tal como se puede observar en la Figura 6.6. En lenguaje coloquial, se podría resumir en "Se encontró el Título" o "No se encontró el Título".

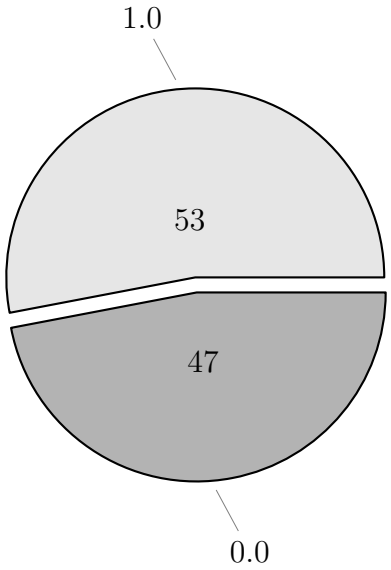

Figura 6.6 – *# de Archivos agrupados por rango de Cobertura obtenido para el Título*

A su vez, se tuvo en cuenta la posibilidad en la cual el título extraído correspondiera parcialmente al título del documento y para dichos casos se consideró un valor de referencia de 0.5. Se buscó reflejar que los resultados obtenidos, si bien no fueron exactos, sí aportan dados de valor al usuario. En la Figura 6.7 se observan dichos resultados.

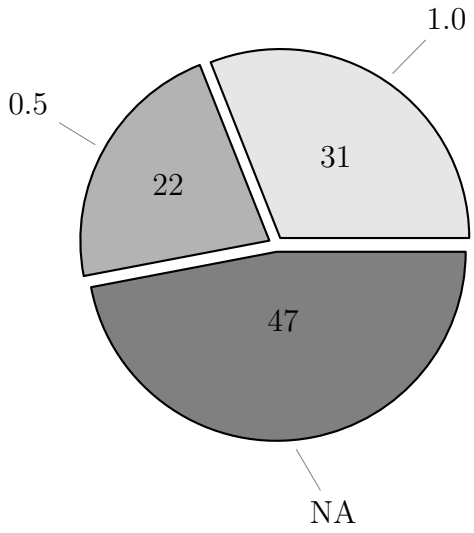

Figura 6.7 – *# de Archivos agrupados por rango de Precisión obtenido para el Título*

Conjunto Documentos en Español:

Se muestra a continuación el valor promedio de Cobertura y Precisión calculado sobre el conjunto de 100 documentos en Español; los valores promedios están calculados para cada una de los elementos principales a extraer en los documentos: *Palabras Claves*, *Autor* y *Título*. Los mismos se muestran en la Tabla 6.2 y Figura 6.8.

	Cobertura	Precisión
Palabras Claves	0.67	0.75
Author	0.45	0.47
Título	0.49	0.44

Tabla 6.2 – *Valores promedio de Cobertura y Precisión para documentos en Español*

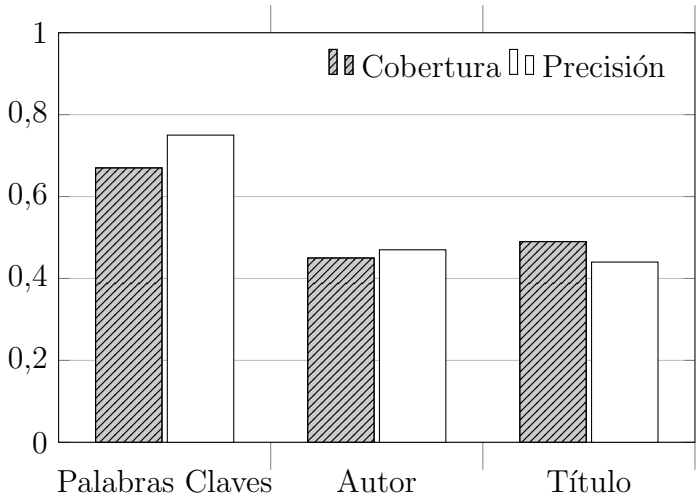

Figura 6.8 – *Valores promedio de Cobertura y Precisión para documentos en Español*

El detalle de los valores obtenidos, segmentados por intervalos, tanto para la Cobertura, como para la Precisión y agrupando por cantidad de archivos de cada segmento se detallan a continuación.

Palabras Claves:

En la Figura 6.9 se puede observar que hay 56 documentos cuya cobertura supera el 75 %. Si comparamos este resultado con los obtenidos para el conjunto de documentos en Inglés, vemos una reducción en la cantidad de información obtenida. Si observamos los resultados para la Precisión, en la Figura 6.10, también se observa una reducción en los valores obtenidos.

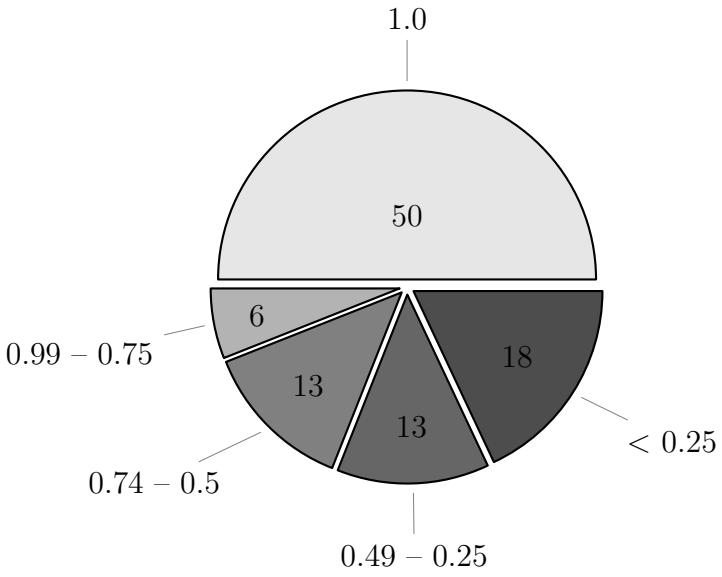

Figura 6.9 – *# de Archivos agrupados por rango de Cobertura obtenido para Palabras Clave*

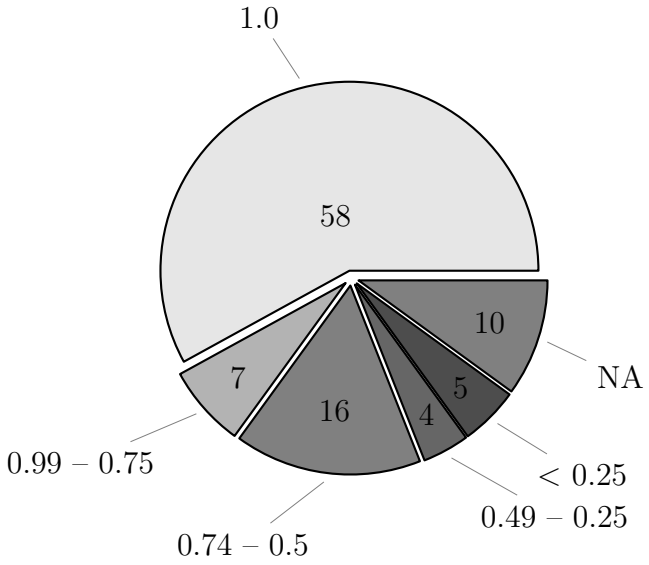

Figura 6.10 – *# de Archivos agrupados por rango de Precisión obtenido para Palabras Clave*

Autor:

Como se puede observar en la Figura 6.11, para la extracción de autores, se obtuvo una cobertura con valor de 1.0 para el 44 % de los documentos evaluados.

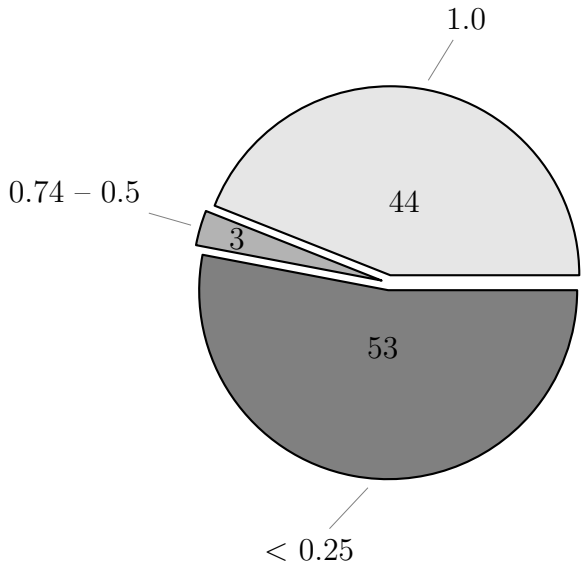

Figura 6.11 – *# de Archivos agrupados por rango de Cobertura obtenido para Autor*

En el caso de la Precisión, Figura 6.12, para el mismo conjunto de documentos, se obtuvieron valores de 1.0 para el 47 % de los archivos. En el 45 % de los casos no se extrajeron resultados, con lo cual no se procedió a evaluar la precisión; estos casos se etiquetan con el valor NA.

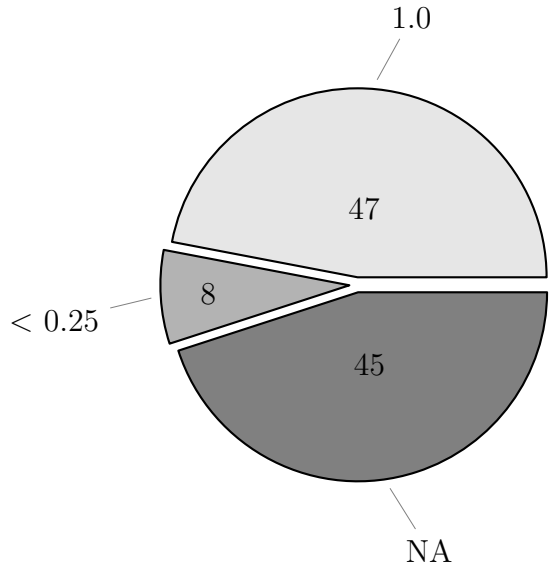

Figura 6.12 – *# de Archivos agrupados por rango de Precisión obtenido para Autor*

Título:

En el caso de la extracción del Título de los documentos, Figuras 6.13 y 6.14, se obtuvieron resultados similares en cuanto a la cobertura que el conjunto de documentos en Inglés (Sección Título).

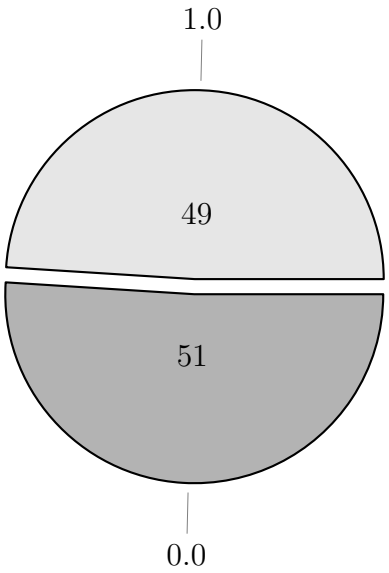

Figura 6.13 – *# de Archivos agrupados por rango de Cobertura obtenido para elTítulo*

Respecto a la Precisión, se puede observar un incremento en los valores obtenidos, con un 41 % de documentos con precisión 1.0, comparados contra un 31 % del conjunto de documentos en Inglés (Sección Título).

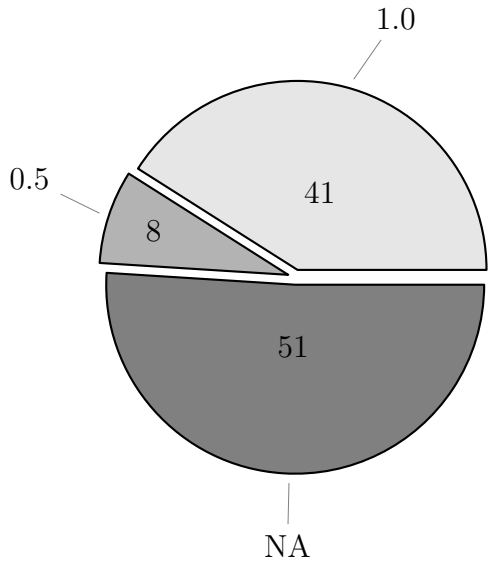

Figura 6.14 – *# de Archivos agrupados por rango de Precisión obtenido para el Título*

Análisis de resultados:

A partir de las primeras pruebas realizadas sobre los conjuntos de documentos (100 en idioma Inglés, 100 en idioma Español), se destaca la gran variedad de documentos que se encontraron en el repositorio, tanto en formato (txt, doc, pdf, pptx) como en su estructura (múltiples columnas, páginas HTML exportadas a PDF, artículos de revistas, etc).

La cobertura promedio fue del 69 % para las Palabras Claves y del 48 % para los Autores y Títulos. De los archivos en los cuales se obtuvo información, la "Precisión" para las Palabras Claves es en promedio del 74 %. En el caso de los Autores y Títulos, la "Precisión" fue el 45 %. En la Tabla 6.3 se listan los resultados obtenidos tanto para documentos en idioma Inglés como Español.

	Cobertura		Precisión	
	Inglés	Español	Inglés	Español
Palabras Claves	0.72	0.67	0.73	0.75
Autor	0.48	0.45	0.51	0.47
Título	0.53	0.49	0.58	0.44

Tabla 6.3 – *Comparación de resultados obtenidos para documentos en Inglés y Español*

Cuando se procedió a analizar el desglose de los documentos por rangos de Precisión y Cobertura , se pudo observar que para ciertos grupos de archivos,

para ambas métricas sus valores superaban el 85 % promedio para los valores extraídos (Palabras Claves, Títulos y Autores) obteniendo para estos grupos excelentes resultados. Haciendo un análisis más detallado, se pudo determinar que la estructura interna de los documentos influye de forma directa en la efectividad de los resultados.

Muchos de los documentos analizados en la primera etapa resultan ser páginas web y/o documentos en formato Word que fueron exportados a PDF, con lo cual su estructura subyacente no puede ser analizada de la misma forma que aquellos documentos que siguen un formato como el de un artículo y/o tesis. En la Figura 6.15 se muestra un ejemplo de un documento cuya estructura no sigue el formato esperado para un artículo y/o tesis tradicional. En la Figura 6.16 se muestra un ejemplo de un documento con la estructura que permite obtener resultados óptimos durante el proceso de extracción.

Figura 6.15 – *Ejemplo de documento con estructura no tradicional*

A partir de este análisis, se procedió a realizar una segunda etapa de pruebas, pero ajustando el conjunto de documentos a aquellos que tenían una estructura

Concept of 'subject' in the context of library and information science from a new angle

Bidyarthi Dutta[a] and Chaitali Dutta[b]

[a]Assistant Professor, Department of Library & Information Science,Vidyasagar University, Midnapore, West Bengal
E-mail: bidyarthi.bhaswati@gmail.com
[b]Associate Professor, Department of Library & Information Science,Jadavpur University, Kolkata

E-mail: contactcdhere@yahoo.com

Received: 02 August 2012, revised 22 May 2013

The concept of subject as expounded in library and information science (LIS) has been interpreted here from the standpoint of the concept of word in linguistics. Both the concepts have been thoroughly reviewed. It has been observed that the concept of subject so long conceived by different researchers in LIS is basically preceded by the concept of document. The description of subject, therefore in most cases, by default becomes incumbent within the concept of document. Since the document is a macroscopic entity, therefore document-dependent description of subject naturally portrays a macroscopic layout of the same. This paper attempts to develop a document-independent description of subject, which is based on semantically-related words within the domain of appropriate context. According to this new description, the subject would eventually become definable as sets of well-defined and semantically-related words that may be regarded as microscopic description. It has also been found out that the seed of document-independent and word-based definition of subject was already sown in the concept of semantic field, a domain under the subject linguistics. This concept was incepted by Trier and subsequently modified by Lehrer. It has been logically established that the idea of foci incepted by Ranganathan and the idea of semantic field incepted and modified by Trier and Lehrer respectively are conceptually equivalent. *A subject* may therefore be described as sets of semantic fields and, in turn as sets of words.

Keywords: Linguistic interpretation of subject, Semantic field, Semantics, Foci, Facet, Macroscopic subject, Microscopic subject, Linguistics, Universe of subjects

Library science and information science: an introduction

Library science is an interdisciplinary or multidisciplinary area of study that deals with collection, processing, organization, preservation, and dissemination of different types of information resources in various kinds of libraries and the enabling of optimum utilization of information by information clientele. Various practical perspectives of different types of academic and research activities come under the purview of this area of study. Traditional libraries usually functioned with mere paper-based, printed materials as information resources, whereas modern concept of libraries embrace wide spectrum of electronic, non-print materials also within the scope of library systems and services. It is interesting to note that

contemporary to the First World War. Thus library is old but library science is new, and library science education is newer. In India, LIS education was started by Borden and Dickinson with the encouragement of Maharaja Swaji Rao of Baroda in 1911, i.e. little more than one hundred years back. The first American school for library science was founded by Melvil Dewey at Columbia University in 1887 and the first textbook on library science was published in the year 1808[1]. The school of thought of library science in India was initiated by the scholar of mathematics, Dr. S.R. Ranganathan, who is known as the Father of library science in India.

Information science, according to Borko[2], "is that discipline that investigates the properties and behavior of information, the forces governing the flow of information, and the means of processing information

Figura 6.16 – *Ejemplo de documento con estructura tradicional*

subyacente de tesis y/o artículo únicamente y se procedió a descartar otros tipos de documentos.

6.3. Resultados segunda fase de pruebas

En base a los resultados obtenidos durante la primera etapa de experimentación, se redujo el conjunto de documentos a aquellos que tuvieran un formato de Tesis y/o Artículo tradicional.

De los conjuntos originales de documentos, se seleccionaron 25 para cada idioma y se procedió a repetir el proceso de extracción para estos dos nuevos conjuntos. A continuación se muestran los resultados obtenidos para esta segunda etapa.

Conjunto Documentos en Inglés:

Se muestra a continuación el valor promedio de Cobertura y Precisión calculado sobre el nuevo conjunto de 25 documentos en Inglés. Los valores promedios están calculados para cada una de los elementos principales a extraer en los documentos: *Palabras Claves*, *Autor* y *Título*. Los mismos se muestran en la Tabla 6.4 y Figura 6.17.

	Cobertura	Precisión
Palabras Claves	0.92	0.93
Author	0.80	0.99
Título	0.83	0.88

Tabla 6.4 – *Valores promedio de Cobertura y Precisión para documentos en Inglés*

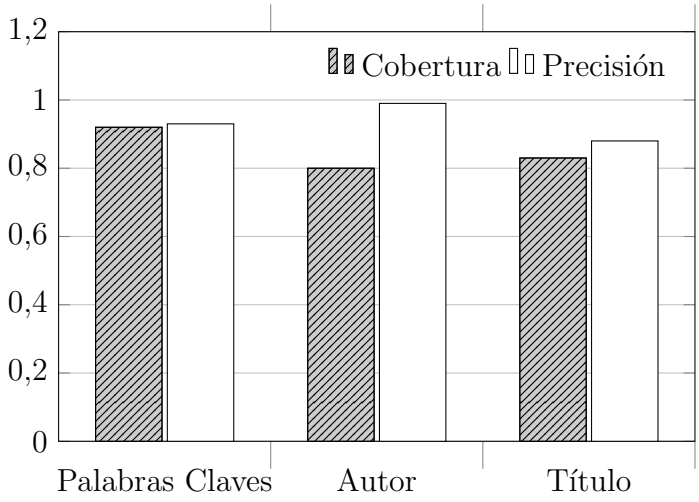

Figura 6.17 – *Valores promedio de Cobertura y Precisión para documentos en Inglés*

El detalle de los valores obtenidos, segmentados por intervalos, tanto para la Cobertura, como para la Precisión y agrupando por cantidad de archivos de cada segmento se detallan a continuación.

Palabras Claves:

En la Figura 6.18, se puede observar como el 96 % de los documentos del conjunto de pruebas tiene valores de cobertura superiores al 0.75, lo cual es un excelente resultado.

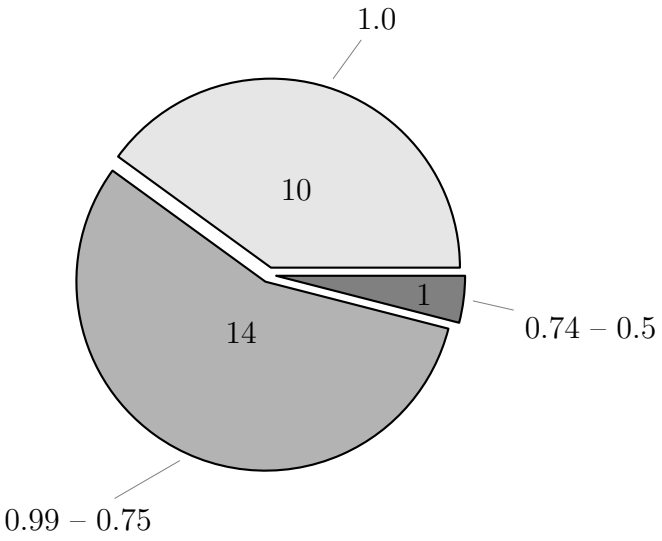

Figura 6.18 – *# de Archivos agrupados por rango de Cobertura obtenido para Palabras Clave*

A su vez para el 100 % de los documentos, Figura 6.19, se obtuvo una Precisión superior al 75 %, lo cual nos indica que se extrajeron datos de valor para todos los documentos evaluados.

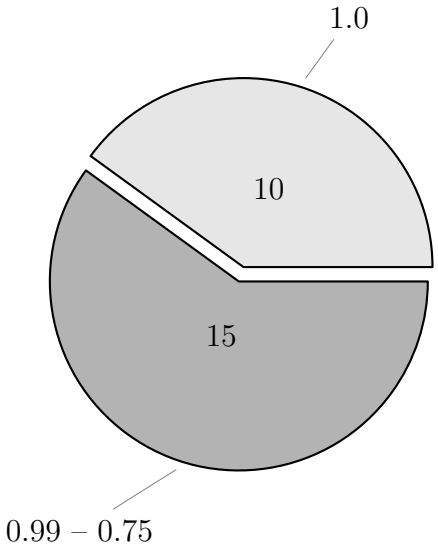

Figura 6.19 – *# de Archivos agrupados por rango de Precisión obtenido para Palabras Clave*

Autor:

Con respecto a los resultados obtenidos para la extracción de Autores, tanto para la Cobertura, como para la Precisión, se obtuvieron muy buenos resultados para el 81 % de los documentos con valores superiores a 0.75, tal cual se puede observa en la Figura 6.20 y la Figura 6.21.

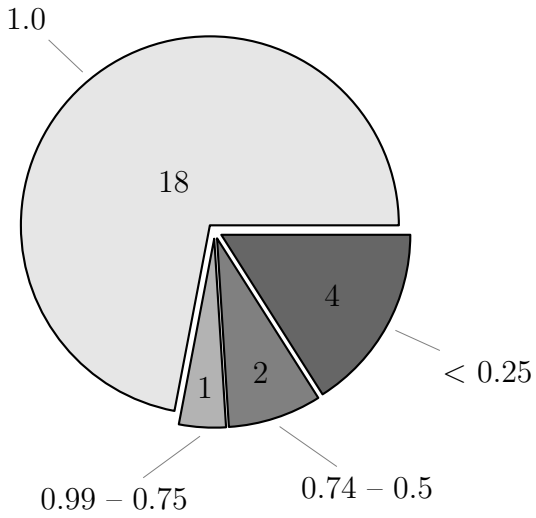

Figura 6.20 – *# de Archivos agrupados por rango de Cobertura obtenido para Autor*

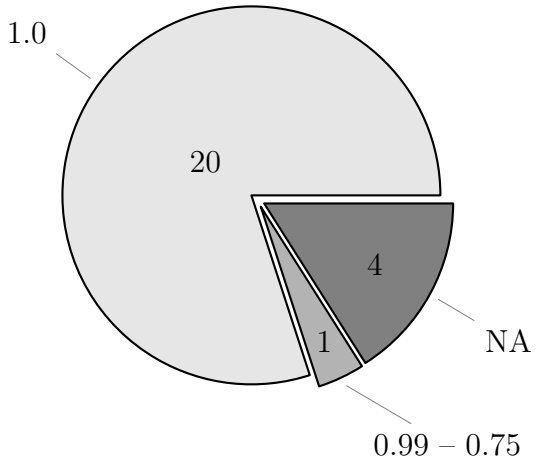

Figura 6.21 – *# de Archivos agrupados por rango de Precisión obtenido para Autor*

Título:

Como se mencionó en la Sección **??**, se consideran 2 posibles escenarios respecto al título, "Se encontró" o "No se encontró"; tomando como referencia un valor de 1.0 para el primer caso y un valor de 0.0 para el segundo. En la Figura 6.22, podemos observar que para un 88 % de los documentos evaluados se pudo extraer el título.

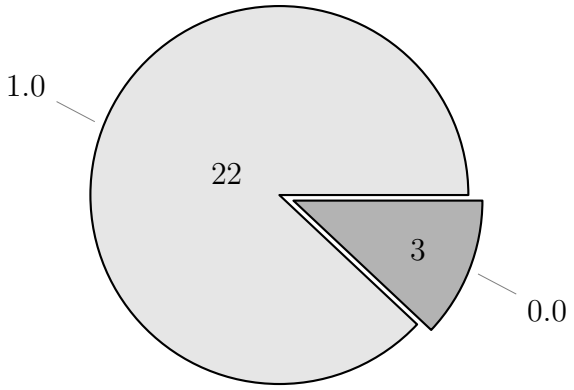

Figura 6.22 – *# de Archivos agrupados por rango de Cobertura obtenido para el Título*

En el caso de la Precisión, en el 52 % de los casos se pudo extraer el título exacto del documento, mientras que en el 16 % se obtuvo un resultado parcialmente correcto. En el 32 % no se produjeron respuestas válidas, por lo cual no se evalúa la precisión correspondiente. Estos valores se observan en la Figura 6.23.

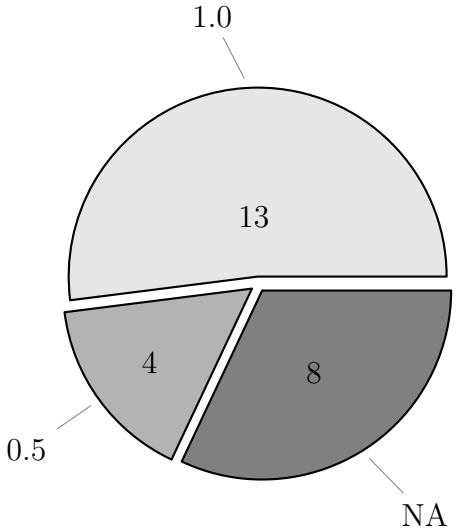

Figura 6.23 – *# de Archivos agrupados por rango de Precisión obtenido para elTítulo*

Conjunto Documentos en Español:

Se muestra a continuación el valor promedio de Cobertura y Precisión calculado sobre el conjunto de 25 documentos en Español; los valores promedios están calculados para cada una de los elementos principales a extraer en los documentos: *Palabras Claves*, *Autor* y *Título*. Los mismos se muestran en la Tabla 6.5 y Figura 6.24.

	Cobertura	Precisión
Palabras Claves	0.98	0.98
Author	0.76	0.80
Título	0.91	0.96

Tabla 6.5 – *Valores promedio de Cobertura y Precisión para documentos en Español*

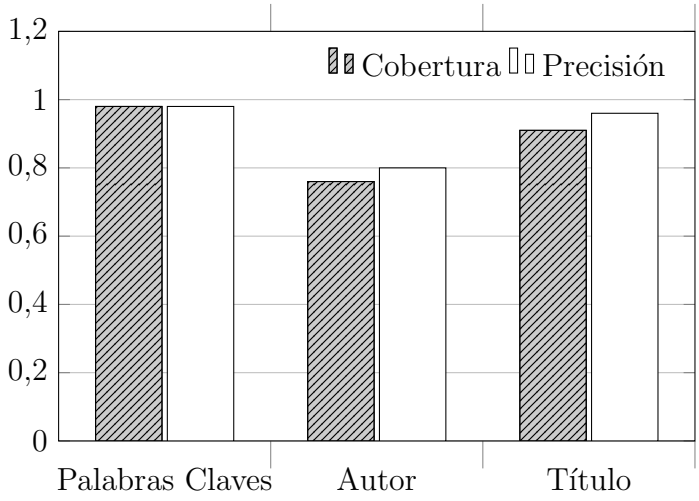

Figura 6.24 – *Valores promedio de Cobertura y Precisión para documentos en Español*

El detalle de los valores obtenidos, segmentados por intervalos, tanto para la Cobertura, como para la Precisión y agrupando por cantidad de archivos de cada segmento se detallan a continuación.

Palabras Claves:

En la Figura 6.18 y Figura 6.26, se puede observar como el 100 % de los documentos del conjunto de pruebas tienen valores de cobertura y precisión superiores al 0.75, lo cual es un excelente resultado.

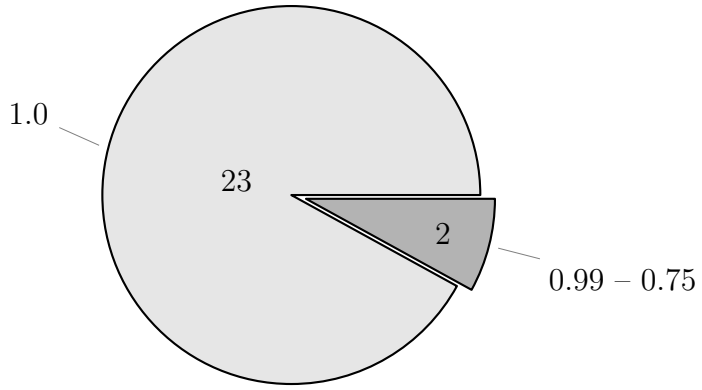

Figura 6.25 – *# de Archivos agrupados por rango de Cobertura obtenido para Palabras Clave*

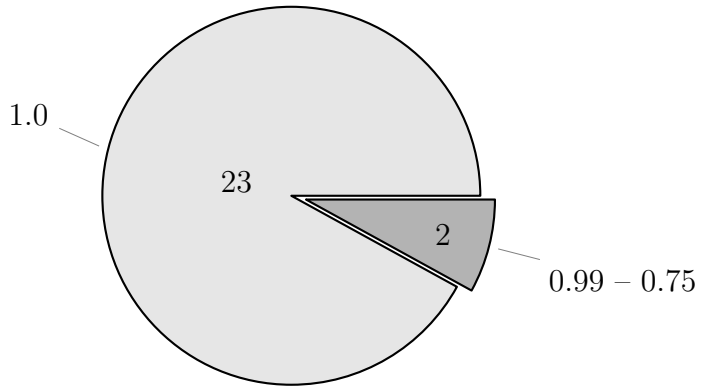

Figura 6.26 – *# de Archivos agrupados por rango de Precisión obtenido para Palabras Clave*

Autor:

Como se puede observar en la Figura 6.27 y en la Figura 6.28 el 80 % de los documentos obtuvieron valores de cobertura y de precisión superiores a 0.75, lo cual se considera un excelente resultado.

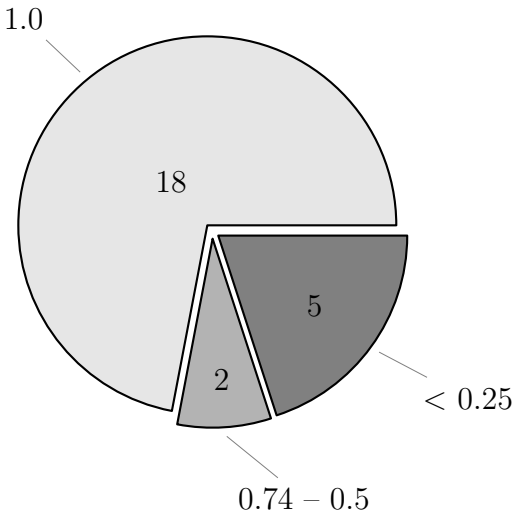

Figura 6.27 – *# de Archivos agrupados por rango de Cobertura obtenido para Autor*

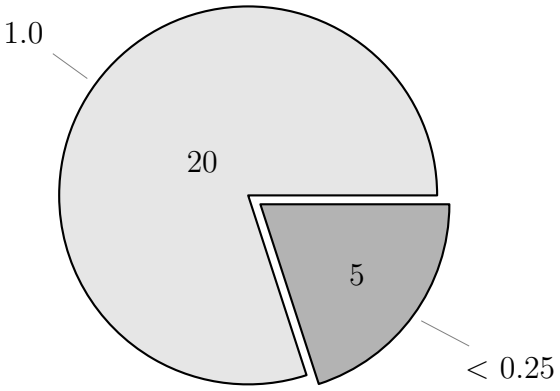

Figura 6.28 – *# de Archivos agrupados por rango de Precisión obtenido para Autor*

Título:

En la Figura 6.29, podemos observar que para un 96 % de los documentos evaluados se pudo extraer el título de forma correcta.

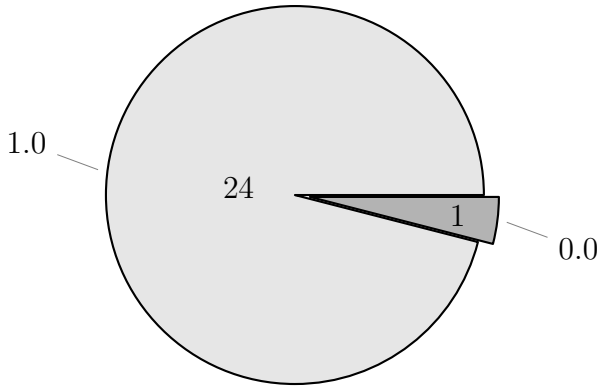

Figura 6.29 – *# de Archivos agrupados por rango de Cobertura obtenido para elTítulo*

En el caso de la Precisión, en el 80 % de los casos se pudo extraer el título exacto del documento, mientras que en el 16 % se obtuvo un resultado parcialmente correcto. En el 4 % de los casos no se produjeron respuestas válidas, por lo cual no se evalúa la precisión correspondiente. Estos valores se observan en la Figura 6.30

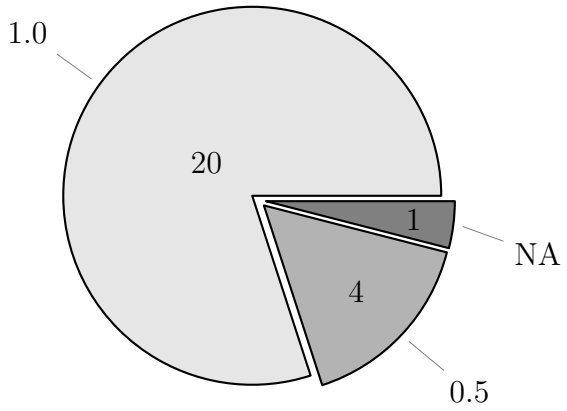

Figura 6.30 – *# de Archivos agrupados por rango de Precisión obtenido para el Título*

Análisis de resultados:

Para esta segunda etapa, con conjuntos de documentos restringidos a 25 por idioma, se pudo determinar:

- La cobertura promedio es superior al 90 % para las palabras claves, y superior al 75 % para los autores y títulos.
- De los archivos en los cuales se obtiene información, la Precisión para las palabras claves es superior al 90 %.

- En el caso de los Autores y Títulos, la Precisión es superior al 80 %.

En la Tabla 6.6 se listan los resultados obtenidos tanto para documentos en idioma Inglés como Español.

	Cobertura		Precisión	
	Inglés	Español	Inglés	Español
Palabras Claves	0.92	0.98	0.93	0.98
Autor	0.80	0.76	0.99	0.80
Título	0.83	0.91	0.88	0.96

Tabla 6.6 – *Comparación de resultados obtenidos para documentos en Inglés y Español - Fase 2*

Estos resultados confirman la importancia de la estructura subyacente del documento al momento de poder automatizar la extracción de metadatos, son claramente los documentos que siguen una estructura de artículo y/o tesis y que tienen un formato tradicional, los que cuentan con mayor tasa de éxito (considerando la cobertura y precisión de la información obtenida).

Capítulo 7

Conclusiones

Para facilitar la carga de objetos digitales educativos en el repositorio se ha modificado el flujo de carga estándar de la plataforma DSpace, presentando un nuevo flujo para el depósito de objetos que permita la incorporación de un extractor de metadatos. Además, se ha propuesto una arquitectura de un Asistente para la extracción automática de algunos metadatos generados de los documentos. Estos metadatos extraídos automáticamente son validados por el usuario en el proceso de descripción del objeto.

Para diseñar el asistente, se analizaron distintas herramientas de extracción y en particular se propuso utilizar la combinación de ParsCit+Alchemy, con la cual se logró incrementar la calidad de los metadatos retornados pasando de un 56 % a un 70 % de efectividad en los resultados obtenidos solo con Alchemy. Antes se desarrolló un prototipo en lenguaje de programación Java del asistente planteado.

Durante una primera etapa de experimentación, se observó que los resultados obtenidos para el conjunto de documentos seleccionados con contenidos y formatos diversos, extraídos de un repositorio presentaban valores promedio de Cobertura y Precisión bajos para poder obtener un proceso de carga completamente automatizado. Se observó que la tasa de error obtenida variaba entre el 30 % y el 50 %. A partir de un análisis más detallado se llegó a la conclusión de que el proceso de extracción era altamente sensible a la estructura subyacente de los documentos a analizar. Se pudo observar que con documentos con estructura tradicional, como puede ser el de tesis y/o articulo en ciertos formatos, se obtenían mejores resultados comparados con otros tipos de documentos, como por ejemplo, páginas HTML exportadas en formato PDF.

Se procedió entonces a realizar una segunda etapa de experimentación, con un subconjunto de documentos que presentaran una estructura homogénea y tu-

vieran un formato adecuado. Los valores obtenidos en esta segunda etapa de experimentación, para las métricas de Cobertura y Precisión fueron considerablemente superiores, lo cual era acorde con la hipótesis planteada anteriormente.

Los resultados obtenidos mediante este prototipo, muestran que la propuesta de incorporar un extractor de metadatos en el proceso de documentos en repositorios DSpace es viable y puede dar buenos resultados. Con los ajustes necesarios se lo va a implementar en el repositorio RepHip de la Universidad Nacional de Rosario. De esta forma se espera ayudar al usuario en el proceso de carga de objetos digitales educativos, disminuyendo así su trabajo y mejorando la cantidad y calidad de los metadatos cargados.

Como trabajo futuro, se plantea el incorporar nuevos módulos que refinen el proceso de extracción para los distintos tipos de documentos y estructuras subyacentes. Dado que la arquitectura planteada para el asistente de carga es sumamente flexible, esto permite añadir y/o remover nuevos extractores. Se puede pensar cada módulo de extracción como un "filtro", en el cual el resultado obtenido puede ser utilizado como entrada del siguiente, consiguiendo de esta manera metadatos cada vez más depurados y específicos de acuerdo al tipo de documento que se proceda a analizar y las preferencias del usuario.

7.1. Publicaciones

Casali, A. Deco C., Bender C., Fontanarrosa S. "Extracción Automática de Metadatos de Objetos Digitales Educativos." Proceedings Novena Conferencia Latinoamericana de Objetos y Tecnologías de Aprendizaje LACLO 2014, pp 23-29. Manizales, Colombia. Octubre 2014.

Casali, A. Deco C., Bender C., Fontanarrosa S. "Extracción Automática de Metadatos como Soporte para el Autoarchivo de Objetos Digitales en Repositorios". Aceptado para publicación en la Edición Especial de Revista Colombiana de Computación (RCC) 2015.

Apéndice A

Código Fuente

```
import os
import glob
import sys
import AlchemyAPI
import argparse
import logging
import logging.handlers
import nltk
from nltk.corpus import stopwords

from subprocess import Popen, PIPE, STDOUT
from bs4 import BeautifulSoup
from lxml import etree

def removeStopwords( palabras, corpus ):
        tokens = nltk.word_tokenize(palabras)
        filtered = [token for token in tokens if not token
            in stopwords.words(corpus)]
        return " ".join(nltk.Text(filtered))

# Enviroment configuration

PARSCIT_PATH = "./ParsCit/bin"
PARSCIT_COMMAND = PARSCIT_PATH + "/" + "citeExtract.pl -m
    extract_header %s"

ALCHEMY_API_KEY = "api_key.txt"

LOG_FILENAME = './metadataExtractionLog.log'

# Set up a specific logger with our desired output level
errorLogger = logging.getLogger('errorLogger')
errorLogger.setLevel(logging.DEBUG)
```

```
formatter = logging.Formatter("%(asctime)s - %(message)s")

# Add the log message handler to the logger
handler = logging.handlers.RotatingFileHandler(
    LOG_FILENAME, maxBytes=20000, backupCount=5)
handler.setFormatter(formatter)

errorLogger.addHandler(handler)

# Command line Arguments

parser = argparse.ArgumentParser(description='Description
    of your program')
parser.add_argument('-f','--file', help='Path to file to
    be parsed', required=True)
parser.add_argument('-c','--collection', help='Collection
    name for the file', required=True)
args = parser.parse_args()


# Parscit processing
infile = args.file
collection = args.collection

errorLogger.debug("Metadata extraction for %s begings" %
    infile)

SHELL_COMMAND = PARSCIT_COMMAND % infile

errorLogger.debug("Execute Parscit in %s" % infile)
errorLogger.debug(SHELL_COMMAND)

shellOutput = Popen(SHELL_COMMAND, shell=True, stdin=PIPE,
    stdout=PIPE, stderr=STDOUT)
output = shellOutput.communicate()[0]

errorLogger.debug("Parscit finish for file %s" % infile)

# create XML
# root element
dspaceMetadata = etree.Element('dspaceMetadata')

# filename
fileName = etree.Element('fileName')
fileName.text = infile
```

```
dspaceMetadata.append(fileName)

collectionStr = etree.Element('collection')
collectionStr.text = collection
dspaceMetadata.append(collectionStr)


# XML manipulation
xmlSoup = BeautifulSoup(output)
alchemyInputFile = ""

errorLogger.debug("Processing result into xml")

# authors
authors = etree.Element('authors')

# concatenate all the author labels found
for authorStr in xmlSoup.find_all('author'):
        authorXML = etree.Element('author')
        authorXML.text = authorStr.string
        authorXML.attrib["confidence"] = authorStr["
           confidence"]
        authors.append(authorXML)

dspaceMetadata.append(authors)

errorLogger.debug("Author extraction finished")

titles  = etree.Element('titles')
for titleStr in xmlSoup.find_all('title'):
        titleXML = etree.Element('title')
        titleXML.text = titleStr.string
        titleXML.attrib["confidence"] = titleStr["
           confidence"]
        titles.append(titleXML)

dspaceMetadata.append(titles)

# concatenate all the abstract labels found
for abstract in xmlSoup.find_all('abstract'):
        alchemyInputFile += abstract.string.encode('utf
           -8')

#extract keywords if find by parscit

keywords  = etree.Element('keywords')
```

```
for key in xmlSoup.find_all('keyword'):
        keywordsArray = key.string.split(';')
        for keywordStr in keywordsArray:
                keywordXML = etree.Element('keyword')
                keywordXML.text = keywordStr
                keywordXML.attrib["confidence"] = "1";
                keywords.append(keywordXML)

if len(keywords) > 0:
        dspaceMetadata.append(keywords)
else:
        if not alchemyInputFile:
                f = open(infile,"r") #opens file with name
                    of "test.txt"
                alchemyInputFile = f.read()
                f.close()

        alchemyInputFile = removeStopwords(
           alchemyInputFile,'spanish')
        alchemyInputFile = removeStopwords(
           alchemyInputFile,'english')

        errorLogger.debug("Abstracts extraction finished")

        # Alchemy Processing

        errorLogger.debug("Beging AlcheyAPI configuration
           ")
        # Create an AlchemyAPI object.
        alchemyObj = AlchemyAPI.AlchemyAPI()

        # Load the API key from disk.
        alchemyObj.loadAPIKey(ALCHEMY_API_KEY);

        errorLogger.debug("Alchemy API Keyword extraction
           beging")

        rankedKeywordsStr = alchemyObj.
           TextGetRankedKeywords(alchemyInputFile)

        errorLogger.debug("Alchemy API Keyword extraction
           finished")

        errorLogger.debug("Alchemy API category extraction
            beging")
```

```
        resultStrCategory = alchemyObj.TextGetCategory(
            alchemyInputFile)

        errorLogger.debug("Alchemy API category extraction
            finished")

        # Generate output
        errorLogger.debug("Metadata XML generation
            begining")

        alchemyCategorySoup = BeautifulSoup(
            resultStrCategory)

        if alchemyCategorySoup.status.string == 'OK':
                language  = etree.Element('language')
                language.text = alchemyCategorySoup.
                    language.string
                category = etree.Element('category')
                category.text = alchemyCategorySoup.
                    category.string
                dspaceMetadata.append(language)
                dspaceMetadata.append(category)

        alchemyKeyWordsSoup = BeautifulSoup(
            rankedKeywordsStr)

        if alchemyKeyWordsSoup.status.string == 'OK':
                keywords  = etree.Element('keywords')
                for keywordStr in alchemyKeyWordsSoup.
                    find_all('keyword'):
                        keywordXML = etree.Element('
                            keyword')
                        keywordXML.text = keywordStr.
                            contents[1].string
                        keywordXML.attrib["confidence"] =
                            keywordStr.relevance.string
                        keywords.append(keywordXML)

        dspaceMetadata.append(keywords)

errorLogger.debug("Metadata XML generation finished")

# pretty string
s = etree.tostring(dspaceMetadata, pretty_print=True,
    encoding="utf-8",xml_declaration=True)
```

```
print s

errorLogger.debug("Metadata generation finished")
```

Bibliografía

[Alfano et al., 2007] Alfano, M., Lenzitti, B., and Visalli, N. (2007). Saxef: A system for automatic extraction of e-learning object features. *Journal of e-learning and Knowledge Society*, 3(2):83–92.

[Beel et al., 2011] Beel, J., Gipp, B., Langer, S., Genzmehr, M., Wilde, E., Nürnberger, A., and Pitman, J. (2011). Introducing mr. dlib, a machine-readable digital library. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 463–464. ACM.

[Casali et al., 2011] Casali, A., Deco, C., Bender, C., and V, G. (2011). Recommender system personalized retrieval of learning objects. In Santos, O. C. and Boticario, J. G., editors, *Educational Recommender Systems and Technologies: Practices and Challenges*, pages 182–210. IGI Global.

[Councill et al., 2008] Councill, I. G., Giles, C. L., and Kan, M.-Y. (2008). Parscit: an open-source crf reference string parsing package. In *LREC*.

[Deng and Yu, 2014] Deng, L. and Yu, D. (2014). Deep learning: Methods and applications. *Found. Trends Signal Process.*, 7(3–4):197–387.

[Duval, 2002] Duval, E. (2002). 1484.12. 1: Ieee standard for learning object metadata. *IEEE LTSC*.

[Li et al., 2005] Li, Y., Dorai, C., and Farrell, R. (2005). Creating magic: system for generating learning object metadata for instructional content. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 367–370. ACM.

[Motz et al., 2009] Motz, R., Badell, C., Barrosa, M., Sum, R., Díaz, G., and Castro, M. (2009). Looking4lo: Sistema informático para la extracción automática de objetos de aprendizaje: Caso de estudio. *IEEE-RITA*, 4(3):223–229.

[Pire et al., 2011] Pire, T., Espinase, B., Casali, A., and Deco, C. (2011). Automatic extraction of learning objects metadata for recommendation: A comparative study. In *XIV Congreso Internacional de Informática en la Educación*.

[San Martín et al., 2013] San Martín, P. S., Bongiovani, P. C., Casali, A., and Deco, C. (2013). Socio-technological perspectives for open access repositories development in the context of public universities in the central-eastern argentina. In *PKP Scholarly Publishing Conference 2013*.

[Smith et al., 2003] Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R., and Walker, J. H. (2003). Dspace: An open source dynamic digital repository.

[Sonntag, 2004] Sonntag, M. (2004). Metadata in e-learning applications: Automatic extraction and reuse. *IDIMT-2004. 12th Interdisciplinary Information Management Talks*, pages 219–231.

[Ting, 2010] Ting, K. (2010). Precision and recall. In Sammut, C. and Webb, G., editors, *Encyclopedia of Machine Learning*, pages 781–781. Springer US.

[Wilbur and Sirotkin, 1992] Wilbur, W. J. and Sirotkin, K. (1992). The automatic identification of stop words. *Journal of information science*, 18(1):45–55.

[Wiley, 2003] Wiley, D. A. (2003). *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.*

[Witten et al., 1999] Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. (1999). Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255. ACM.

[Yuen, 2007] Yuen, T. W. (2007). Automatic extraction of learning object metadata (lom) from html web pages. *Master of philosophy, City University of Hong Kong.*