INSTITUTO DE INGENIERÍA UNAM

Desarrollo de un sistema de publicación de Biblioteca de arte mexicano

DspaceLoader+TopicModeling

Carlos Emiliano González Gallardo

Carlos Francisco Méndez Cruz

Gerardo Sierra Martínez

Azucena Montes Rendón

3 de julio de 2015

Circuito Escolar, Ciudad Universitaria
Delegación Coyoacán, CP 04510, México, DF
www.ii.unam.mx

Contenido

1. Información básica

DSpaceLoader+TopicModeling (DSL+TM) es un software a la medida desarrollado por el Grupo de Ingeniería Lingüística del Instituto de Ingeniería de la UNAM (GIL) para la empresa OCRMX. DSL+TM se encuentra integrado por dos grandes módulos: DSpaceLoaderV2 y TopicModelingV1.1. El primero está encargado de cargar e indexar documentos a DSpace de forma automática y el segundo está encargado de crear un modelo de similitud semántica entre los documentos cargados a DSpace.

DSL+TM está diseñado para correr en sistemas Unix o similares, ha sido probado en distribuciones Linux (Ubuntu y OpenSuse) así como en Apple OS X Yosemite.

2. Estructura de DSL+TM

DSL+TM contiene un shellscript (DSpaceLoader+TopicModeling.sh) que hace el llamado a los módulos DSpaceLoaderV2 y TopicModelingV1.1. Dentro del shellscript de DSL+TM se encuentran definidos 11 parámetros obligatorios para su correcto funcionamiento. A continuación se muestra su código.

```
#!/bin/bash
#DSpaceLoader+TopicModeling.sh
#########
#     31/08/2015
#     Sistema desarrollado por el GIL, Instituto de Ingenieria UNAM
#     cgonzalezg@iingen.unam.mx
#########

#NOTA: La entrada al sistema debe ser única y exclusivamente a través de este
script
CANTIDAD_LIBROS="500"
RUTA_WAREHOUSE="/export/librarian/"
NOMBRE_XLSX="/home/dspaceadmin/DSpaceLoader+TopicModeling/inventarioNvo.xlsx
"
RUTA_GENERAL_LIBROS="/export/dspacearchive/libros"
EPERSON="carlos.fuentes@orcmx.com"
RUTA_BASE_TXTS="/export/dspacearchive/txt"
```

```
RUTA_LSI="/export/dspacearchive/lsi"
RUTA_XML="/home/dspaceadmin/dspace/webapps/xmlui/themes/Mirage/lib/xsl/aspec
t/artifactbrowser"
BD_USUARIO="dspace"
BD_PASSWORD="4)2;V>X"
URL_BASE="http://200.66.82.41:8080"
./DSpaceLoader2/DSpaceLoader.sh $CANTIDAD_LIBROS $RUTA_WAREHOUSE $NOMBRE_XLSX
$RUTA_GENERAL_LIBROS $EPERSON
./TopicModeling1.1/TopicModeling.sh   $RUTA_GENERAL_LIBROS   $RUTA_BASE_TXTS
$RUTA_LSI $RUTA_XML $BD_USUARIO $BD_PASSWORD $URL_BASE
```

Junto con DSpaceLoader+TopicModeling.sh se encuentra un archivo informativo (README.txt) y los dos módulos: DSpaceLoaderV2 y TopicModelingV1.1 (ver Figura 2.1).

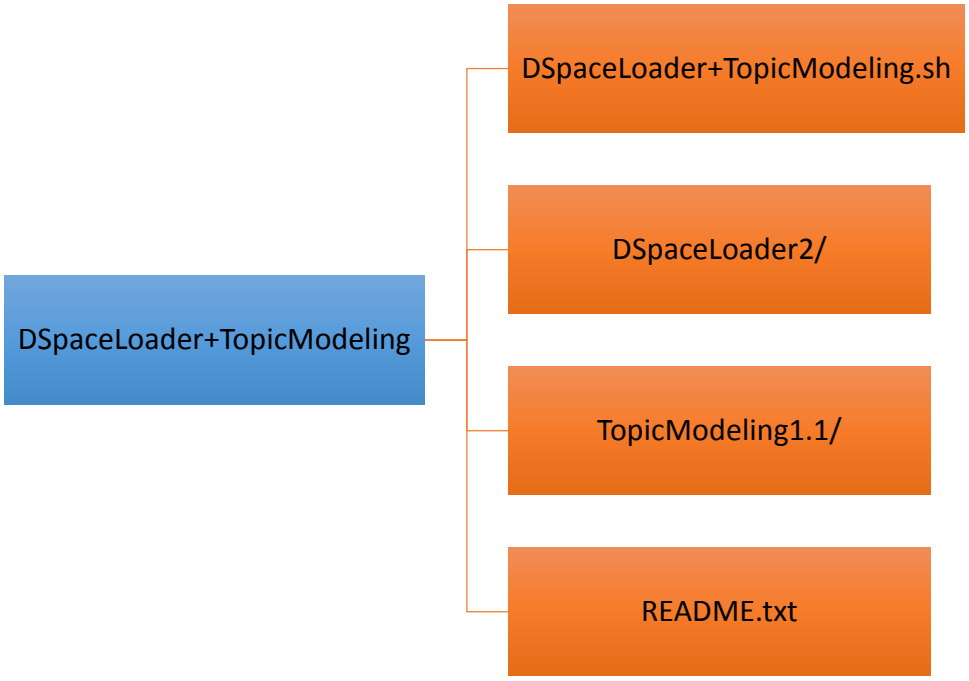


Figura 2.1 Estructura de DSL+TM

2.1. Parámetros de DSL+TM

DSL+TM trabaja con 11 parámetros que son redireccionados a sus módulos: DSpaceLoaderV2 y TopicModelingV1.1. Es obligatorio definir estos parámetros dentro del script para un correcto funcionamiento de DSL+TM.

A continuación se hace la descripción de cada uno de los 11 parámetros:

- `CANTIDAD_LIBROS`: Cantidad de libros que se intentarán cargar a DSpace y que serán procesados por TopicModelingV1.1. Este párametro es absoluto, lo que significa que si en alguna ejecución anterior de DSL+TM se le asignó un valor de 100 (100 libros) y en una ejecución subsecuente se desea tener un total de 200 libros dentro de DSpace, el valor a establecer sería 200 en lugar de 100.

- `RUTA_WAREHOUSE`: Carpeta o dirección del warehouse o almacén en donde se encuentran los libros en .pdf con la estructura que se estableció en un principio. En la sección **DSpaceLoaderV2** se detallará esta estructura.

- `NOMBRE_XLSX`: Archivo excel .xlsx de metadatos. En la sección **DSpaceLoaderV2** se detallará la estructura de este archivo.

- `RUTA_GENERAL_LIBROS`: Carpeta en donde se colocarán los libros agrupados en porciones de aproximadamente 50Mb.

- `EPERSON`: Eperson que será utilizado para hacer la carga de libros. Este usuario debe existir peviamente en DSpace.

- `RUTA_BASE_TXTS`: Carpeta en donde se colocarán los libros completos en .txt. En la sección **TopicModelingV1.1** se detallará la estructura de esta carpeta.

- `RUTA_LSI`: Carpeta en donde se almacenarán las estructuras del modelo LSI para similitud semántica entre documentos. En la sección **TopicModelingV1.1** se detallará el contenido de esta carpeta.

- `RUTA_XML`: Carpeta en donde se colocará el archivo xml de similitud semántica entre documentos. En la sección **TopicModelingV1.1** se detallará la estructura del archivo.

- `BD_USUARIO`: Usuario de la base de datos llamada dspace.

- `BD_PASSWORD`: Password del usuario definido en `BD_USUARIO`.

- `URL_BASE`: Dirección pública y puerto de la instalación de DSpace. Se asume que ya existe una versión de DSpace en línea en esta dirección. Se usa para poner el enlace de los libros similares de cada item de DSpace.

3. Requisitos mínimos del sistema

DSpaceLoader+TopicModeling está compuesto por un conjunto de scripts de Python, por lo que el intérprete de Pyton 2.7 debe estar instalado en el sistema.

El uso del procesador y memoria RAM así como de Disco Duro depende totalmente de la cantidad de libros a procesar.

4. Dependencias

DSpaceLoader+TopicModeling no utilza bibliotecas externas. Favor de revisar los archivos README.txt de DSpaceLoaderV2 y TopicModelingV1.1 para ver las dependencias particulares.

5. Instalación

1. Descomprimir el archivo DSpaceLoader+TopicModeling.zip en la carpeta deseada.
2. Asegurarse de tener instalado easy_install en el sistema, en caso de que no esté instalado es posible hacerlo ejecutando la siguiente instrucción:

```
sudo wget https://bootstrap.pypa.io/ez_setup.py -O - | python
```

3. Utilizar easy_install para descargar e instalar los siguientes paquetes:
   a. openpyxl

```
Carloss-MacBook-Air:~ cicak$ sudo easy_install openpyxl
Password:
```

   b. pyPdf2

```
Carloss-MacBook-Air:~ cicak$ sudo easy_install pyPdf2
```

   c. pdfminer

```
Carloss-MacBook-Air:~ cicak$ sudo easy_install pdfminer
```

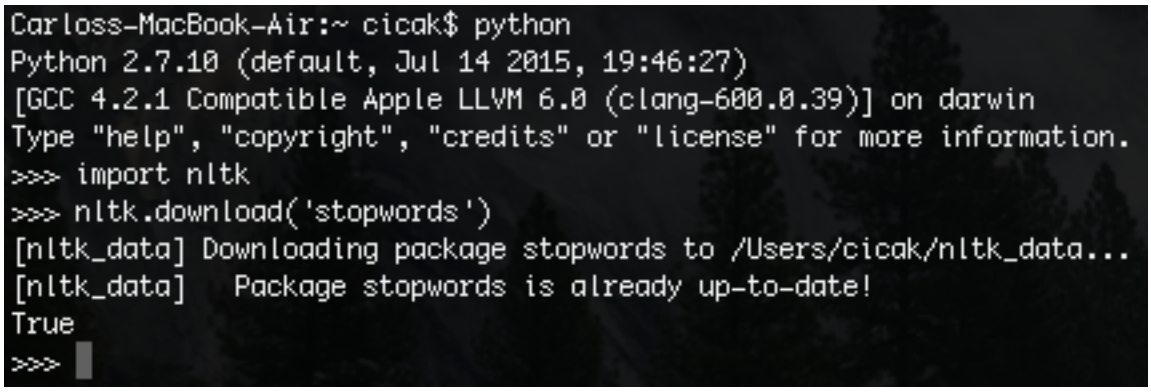
   d. gensim

```
Carloss-MacBook-Air:~ cicak$ sudo easy_install gensim
```

   e. nltk

```
Carloss-MacBook-Air:~ cicak$ sudo easy_install nltk
```

4. Descargar stopwords utilizando nltk



6. Ejecución de DSL+TM

La ejecución de DSpaceLoader+TopicModeling se hace mediante el script llamado "DSpaceLoader+TopicModeling.sh". Es necesario que este cuente con permisos de ejecución.

NOTA IMPORTANTE: ES IMPORTANTE EJECUTAR DSpaceLoader+TopicModeling CON PERMISOS DE SUPERUSUARIO, DE LO CONTRARIO NO SERÁ POSIBLE INTERACTUAR CON DSpace Y SE PRESENTARÁN ERRORES.

DSpaceLoader+TopicModeling no recibe parámetros por línea de comandos. Los parámetros se encuentran definidos dentro del mismo script "DSpaceLoader+TopicModeling.sh". Estos parámetros son enviados a DSpaceLoaderV2 y TopicModelingV1.1. La forma de ejecutar DSpaceLoader+TopicModeling es:

```
sudo ./DSpaceLoader+TopicModeling.sh
```

7. DSpaceLoaderV2

7.1. Información básica

DSpaceLoaderV2 es un software a la medida desarrollado por el GIL para la empresa OCRMX. DSpaceLoaderV2 tiene como finalidad extraer la información localizada en un sistema de archivos para estructurarla en un formato útil para DSpace.

DSpaceLoaderV2 está diseñado para correr en sistemas Unix o similares, ha sido probado en distribuciones Linux (Ubuntu y OpenSuse) así como en Apple OS X Yosemite. DSpaceLoaderV2 busca por cada libro de la carpeta base indicada (RUTA_WAREHOUSE) una carpeta llamada "pdf", de la cual extrae las hojas escaneadas, y una carpeta "jpg", de la cual extrae la imagen de portada del libro. Por esto es necesario respetar el formato que se estableció en un inicio (ver Figura 7.1.1).

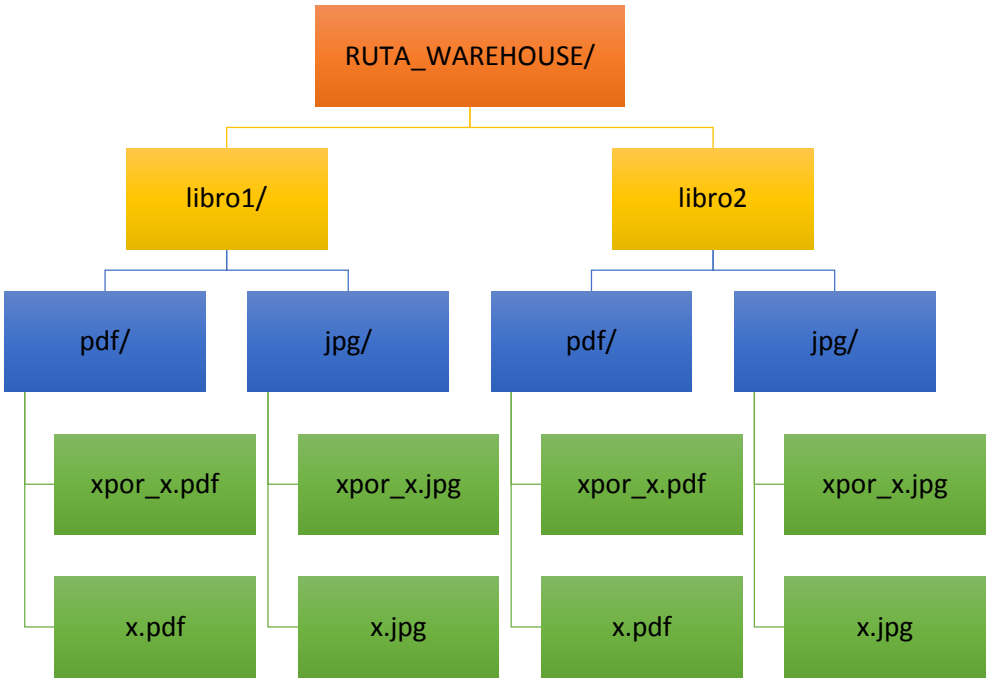


Figura 7.1.1 Formato Warehouse

Cada libro debe contener dos carpetas. La primera, con el nombre de “pdf/”, contiene una lista de archivo (hojas) en formato PDF que componen el libro. El archivo que hace referencia a la portada del libro debe llevar el nombre xxxpor_xxx.pdf, esto es necesario para que al unir las hojas se pueda colocar al inicio la portada del libro. La segunda, con el nombre de “jpg/”, contiene una lista de archivo (hojas) en formato JPG que componen el libro. El archivo que

hacer referencia a la portada del libro debe llevar el nombre xxxpor_xxx.jpg, esto es necesario para poder crear una imagen de vista previa del libro.

El archivo de Excel (NOMBRE_XLSX) que contiene los metadatos de los libros debe contar obligatoriamente con las siguientes columnas:

- **Final**:
  Nombre del libro que debe ser el mismo que la carpeta en donde se encuentran almacenadas las hojas escaneadas (RUTA_WAREHOUSE).
  Valores posibles: [Título del libro]
  Ejemplo: 12_artistas_donde_se_origina_el_arte_en_el_aire

- **Autor**:
  Nombre el autor del libro.
  Valores posibles: [Autor del libro | 0 | #N/A]
  Ejemplo: 0

- **Año de edición**:
  Año en que se publica el libro.
  Valores posibles: [Año de edición| 0 | #N/A]
  Ejemplo: #N/A

- **Colección**:
  Número que corresponde a la colección dentro de DSpace en la cual se desea colocar el libro. Esta colección debe existir previamente en DSpace.
  Valores posibles: [Colección]
  Ejemplo: 3

- **Licencia**:

  Texto que contiene la licencia del libro.

  Valores posibles: [Licencia]

  Ejemplo: Licencia básica

La Figura 7.1.2 muestra un ejemplo del archivo de Excel de metadatos.

ID	Final	Autor	Edición	Año de	Mes de	Págs.	Colecció	Licencia
1	101_masterpieces_of_american_primative_painting	James J. Rc	0	1961	0	159	1	Licencia Básica
2	1200_years_of_italian_sculpture	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
3	12_artistas_donde_se_origina_el_arte_en_el_aire	0	0	2000	0	500	1	Licencia Básica
4	12_dibujos_de_jose_maria_velasco	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
5	20_dibujos_mexicanos_de_maroto	0	0	0	0	30	1	Licencia Básica
6	25_estudios_de_folklore	UNAM	1	1971	0	329	1	Licencia Básica
7	300_anos_de_fraudes_en_el_comercio_de_antiguedades	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
8	330_grabados_originales_manuel_manilla	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
9	45_contemporary_mexican_artists	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
10	50_anos_de_danza_en_el_palacio_de_bellas_artes_1934_-_1984_vol._2	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
11	abraham_angel_y_su_tiempo	Lic. Augustc	0	1985	Marzo-Junic	55	1	Licencia Básica
12	abstract_and_surrealist	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
13	acambaro_colonial	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
14	acapulco	Enrique f. G	1	1959	Diciembre	62	1	Licencia Básica
15	accion_de_las_naciones_unidas_en_mexico	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
16	a_cien_anos_del_5_de_mayo_de_1862	Manuel J. S	1	1962	Abril	527	1	Licencia Básica
17	actualidad_de_bartolome_de_las_casa_1975	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
18	adela_breton	Mario De La	0	1993	Noviembre	147	1	Licencia Básica
19	adriana_yanes_el_movimiento_surrealista	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
20	african_art	Werner Sch	0	1954	0	175	1	Licencia Básica
21	a_grevin_le_monde_amusant	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
22	aguafortistas	0	0	0	0	60	1	Licencia Básica
23	a_guide_to_mexican_art	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
24	agustin_lazo	James Oles	1	2009	Noviembre	207	1	Licencia Básica
25	agustin_lazo_1928	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
26	alameda	Gerardo Est	0	2001	0	287	1	Licencia Básica
27	a_la_politica_en_el_arte	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
28	album_de_hidalgo	0	0	1864	Septiembre	270	1	Licencia Básica
29	album_del_450_aniversario_de_las_apariciones_de_nuestra_senora_de_guadalupe	Hector Vázc	0	1981	0	299	1	Licencia Básica
30	album_de_las_estrellas	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
31	album_del_tiempo_perdido	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
32	album_de_palegrafia_hispanoamericana_de_los_siglos_xvi_y_xvii_vol._iii_1955	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
33	album_la_arquitectura_en_mexico	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica
34	album_mexicano_prudhomme_1843	#N/A	#N/A	#N/A	#N/A	#N/A	1	Licencia Básica

Figura 7.1.2 Ejemplo archivo de metadatos

7.2. Estructura de DSpaceLoaderV2

La ejecución de DSpaceLoaderV2 es por medio del script DSpaceLoader.sh. Este, a su vez, hace el llamado al script de Python scripts/DSpaceLoader.py. Este script es el encargado de llamar al resto de los scripts sin la intervención del usuario.

Dentro de la carpeta mapfiles/ se almacenan los archivos mapfile generados por DSpace. Junto con DSpaceLoader.sh se encuentra un archivo informativo (README.txt), un archivo de log (DSpaceLoader.log) y un archivo de registros (LibrosCargados.dsl). Estos dos últimos archivos son generados por DSpaceLoaderV2 la primera vez que es ejecutado.

Esta organización de archivos puede verse de forma gráfica enseguida (ver Figura 7.2.1).

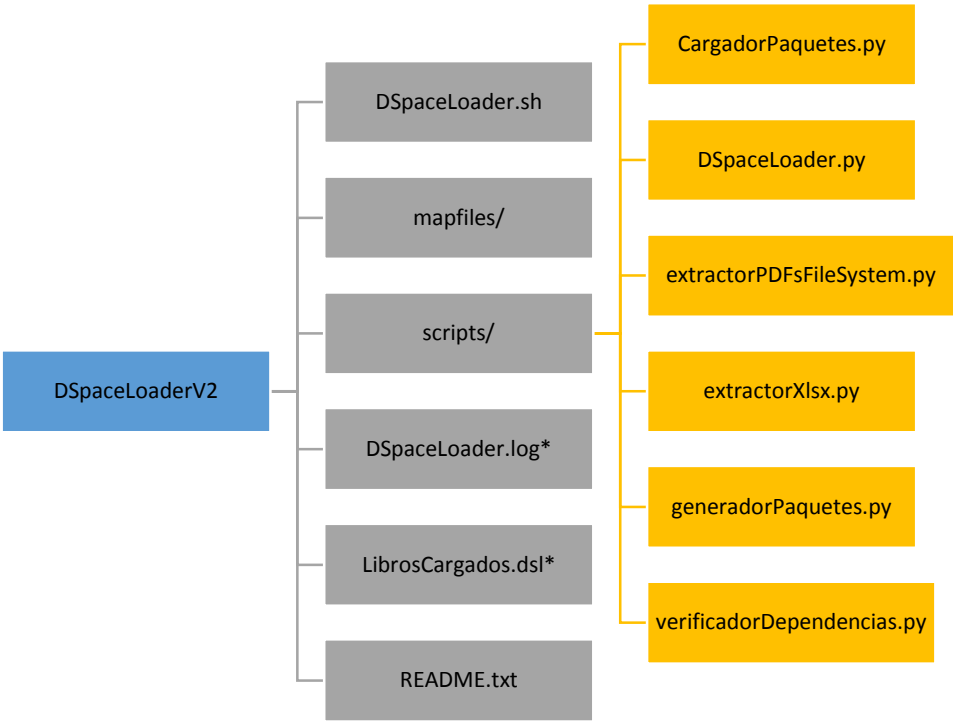


Figura 7.2.1 Estructura de DSpaceLoaderV2

7.3. Flujo de DSpaceLoaderV2

DSpaceLoaderV2 se compone de cinco fases (4-8) del Sistema de publicación de Biblioteca de arte mexicano (ver Figura 7.5.1).

4. **Gathering metadata**: Extrae la información de los libros contenida dentro del archivo de metadatos.
5. **Validation, retrieval & merge (50Mb)**: Por cada libro dento del Warehouse, valida que existan los archivos necesarios para su correcta extracción, extrae las hojas del libro, las agrupa en volúmenes de aproximadamente 50Mb y extrae la imagen de portada del mismo.

6. **Assigning metadata to item directory & creating Simple Archive Format (SAF)**: Por cada libro procesado en 5, se crean los archivos necesarios para ser reconocidos por DSpace; creando así paquetes SAF.
7. **Upload items**: Por cada paquete SAF creado se hace una petición a DSpace para cargarlo a su sistema.
8. **Filter media**: Realiza una petición de indexado a DSpace.

7.4. Requisitos mínimos del sistema

DSpaceLoaderV2 está compuesto por un conjunto de scripts de Python, por lo que el intérprete de Pyton 2.7 debe estar instalado en el sistema. El uso del procesador y memoria RAM así como de Disco Duro depende totalmente de la cantidad de archivos PDF a extraer del servidor.

7.5. Dependencias

DSpaceLoaderV2 hace uso de bibliotecas externas que son de código abierto. Al iniciarse, DSpaceLoaderV2 hace una revisión de las bibliotecas necesarias. En caso de faltar alguna, cancela toda acción posterior. Los paquetes necesarios para el correcto funcionamiento de DSpaceLoaderV2 son:

- openpyxl
  https://openpyxl.readthedocs.org/en/latest/
- pyPdf2
  https://pypi.python.org/pypi/PyPDF2/1.24
- pdfminer
  https://pypi.python.org/pypi/pdfminer/

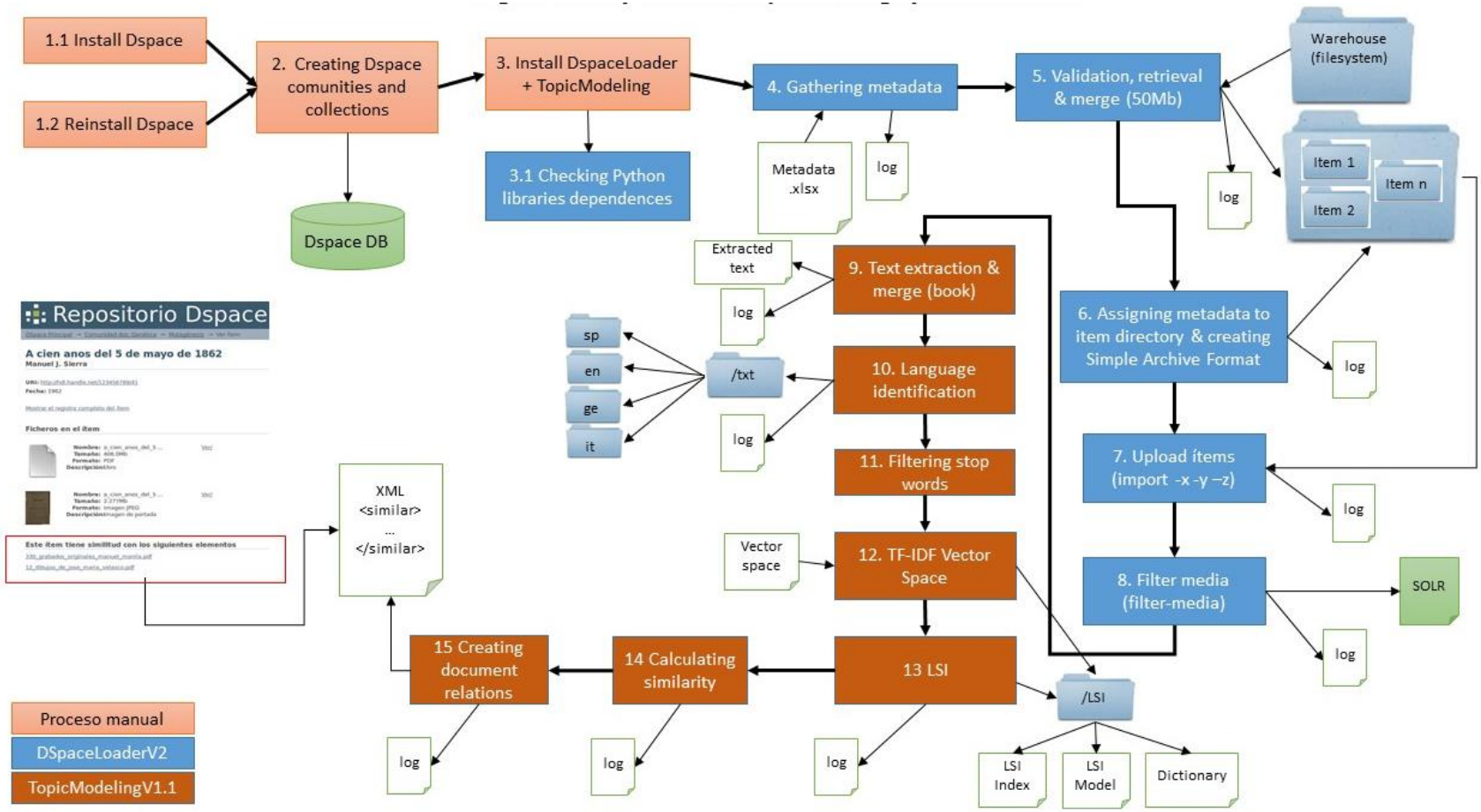


Figura 7.5.1 Arquitectura del Sistema de publicación de Biblioteca de arte mexicano

Es recomendable tener instalado "easy_install" para hacer la instalación de estos paquetes. "easy_install" se encuentra en "setuptools 18.0.1", las instrucciones de descarga e instalación se pueden consultar en la siguiente liga: https://pypi.python.org/pypi/setuptools

Una vez instalado "easy_install", la instalación de los paquetes es muy sencilla:

```
sudo easy_install [openpyxl|pyPdf2|pdfminer]
```

Ejemplo: `sudo easy_install openpyxl`

7.6. Ejecución de DSpaceLoaderV2

DspaceLoaderV2 es un módulo de DSL+TM por lo que no se recomienda su ejecución de forma independiente.

8. TopicModelingV1.1

8.1. Información básica

TopicModelingV1.1 es un software a la medida desarrollado por el GIL para la empresa OCRMX. TopicModelingV1.1 está diseñado para correr en sistemas Unix o similares. Ha sido probado en distribuciones Linux (Ubuntu y OpenSuse) así como en Apple OS X Yosemite.

TopicMOdeling V1.1 extrae el texto de los archivos pdf que se encuentran en la carpeta indicada. Una vez extraido el texto, detecta el idioma del libro y lo almacen en la carpeta de idioma correspondiente. Por cada carpeta de idioma generada, realiza una comparación semántica de los libros obteniendo un diccionario de similitudes. Finalmente escribe un archivo xml con el formato requerido por DSpace en donde muestra aquellos libros con una similitud coseno >= 0.5.

El formato del archivo XML que es creado para mostrar las similitudes entre los libros es el siguente:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<similitud>
    <libro handle=”123456789/#”>
        <similar>
            <titulo>Nombre</titulo>
            <href>ip:puerto/xmlui/handle/123456789/#</href>
        </similar>
    </libro>
    . . .
    <libro handle=”123456789/#”>
        <similar>
            <titulo>Nombre</titulo>
            <href>ip:puerto/xmlui/handle/123456789/#</href>
        </similar>
    </libro>
</similitud>
```

8.2. Estructura de TopicModelingV1.1

La ejecución del TopicModelingV1.1 es por medio del script TopicModeling.sh. Este a su vez hace el llamado al script de python scripts/TopicModeling.py. Este script es el encargado de llamar al resto de los scripts sin la intervención del usuario. Junto con DSpaceLoader.sh se encuentra un archivo informativo (README.txt).

Dentro de la carpeta en donde se almacenan los libros en .txt, el archivo librosAgregados.tm se encarga de llevar un regitro de aquellos libros que ya han sido convertidos. Éste archivo es creado automáticamente la primera vez que TopicModelingV1.1 es ejecutado. Esta organización de archivos y directorios puede verse de forma gráfica en la .

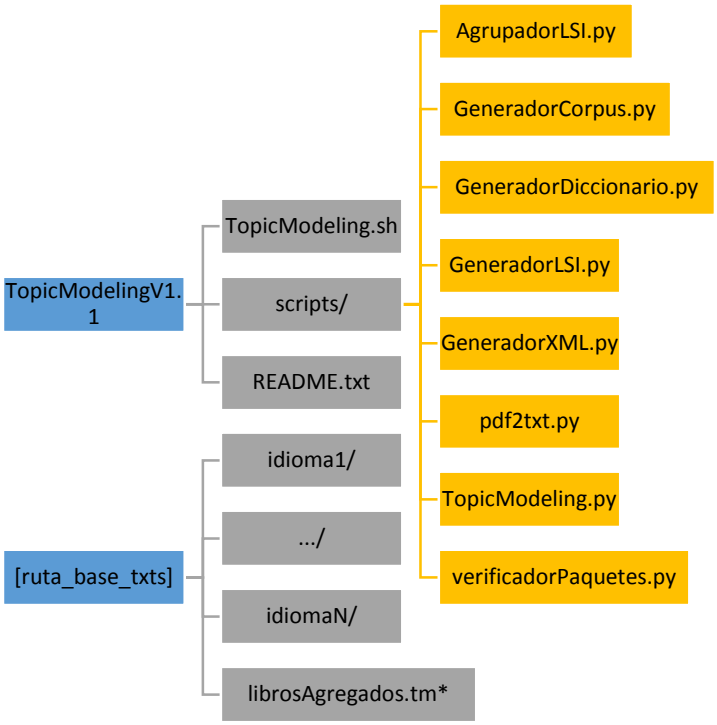


Figura 8.2.1 Estructura de TopicModelingV1.1

8.3. Flujo de TopicModelingV1.1

TopicModelingV1.1 se compone de siete fases (9-15) de la Figura 7.5.1.

9. **Text extraction & Merge (book)**: Cada uno de los libros agrupado por 5 es procesado y su texto es extraido.
10. Language identification: Cada texto de los libros es analizado a partir de stopwords para así identificar su idioma.
11. Filtering stop words: Una vez que se ha identificado el idioma de un texto, sus stopwords son descartadas y finalmente se guarda el archivo en la carpeta del idioma indicado.

12. TF-IDF Vector Space: Por cada libro perteneciente a un idioma se obtiene una representación vectorial del mismo y posteriormente se hace una transformación a TF IDF.
13. LSA: A partir de la tranformación TF-IDF se hace una transformación para representar a cada libro por un grupo de
14. Calculating similarity: Por medio de la distancia conseno se mide la similitud entre los documentos y son seleccinados aquellos con un valor de similitud mayor a 0.5.
15. Creating document relations: La similitudes entre los libros son volcadas en el archivo XML que DSpace utiliza para mostrar los libros similares.

8.4. Requisitos mínimos del sistema

TopicModelingV1.1 está compuesto por un conjunto de scripts de Python; por lo que el intérprete de pyton 2.7 debe estar instalado en el sistema.
El uso del procesador y memoria RAM así como de Disco Duro depende totalmente de la cantidad de archivos PDF a extraer del servidor.

8.5. Dependencias

TopicModelingV1.1 hace uso de bibliotecas externas que son código abierto. Al iniciarse TopicModelingV1.1 hace una revisión de las bibliotecas necesarias, en caso de faltar alguna cancela toda acción posterior.
Paquetes necesarios para el correcto funcionamiento de TopicModelingV1.1:

- gensim
  https://radimrehurek.com/gensim/
- nltk
  http://www.nltk.org/

- pdfminer
  https://pypi.python.org/pypi/pdfminer/

Es recomendable tener instalado "easy_install" para hacer la instalación de estos paquetes. "easy_install" se encuentra en "setuptools 18.0.1", las instrucciones de descarga e instalación se pueden consultar en la siguiente liga: https://pypi.python.org/pypi/setuptools

Una vez instalado "easy_install", la instalación de los paquetes es muy sencilla:

```
sudo easy_install [gensim|nltk|pdfminer]
```

Ejemplo: `sudo easy_install pdfminer`

Una vez instalado el paquete NLTK es necesario descargar unas listas de stopwords necesarias para hacer la identificación automática de idioma.

```
$python
>>> import nltk
>>> nltk.download('stopwords')
[nltk_data] Downloading package stopwords to
/Users/cicak/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
>>>
```

8.6. Ejecución de TopicModelingV1.1

TopicModelingV1.1 es un módulo de DSL+TM por lo que no se recomienda ejecutarlo de forma independiente.