Identification of Science Resources & Extraction of Standard Metadata Properties

Pratik Shrivastava[1], Dave Vieglais[2]

[1]University of Illinois at Urbana-Champaign, University of New Mexico

Introduction

- Identifying the correct file format is imperative for processing its contents.
- Many metadata standards are serialized as XML requires additional details of namespace information for processing.
- Packaging data into data packages requires metadata identification and parsing of the files.
- A tool for reliable identification makes it easier.

Aim

- Determine the scientific resources using the Linux file command and Apache Tika which are excellent tools for file format identification.
- Use Apache Tika for parsing the metadata contents of the resources.
- Extraction of standard set of properties from the metadata.

File Command

- File command performs several additional tests for determining the file format instead of using the file extensions.
- Uses the format signatures, known as magic numbers for identifying the file format.
- The magic directory contains the files, these files consist of the magic numbers. File command uses a compiled binary file containing the magic files.

Apache Tika

- It is an open source toolkit for detecting and extracting metadata and contents of the files.
- Its ability to detect and parse file formats from over a 1000 different formats makes it a useful tool for search engine indexing, content analysis, translation etc.
- The new file types can be detected by creating a custom XML file containing the information.
- New parsers can be easily created and integrated into the application for fresh file formats.

Creating Libmagic file

- Create Magic files for identifying DataONE file formats.
- Gathered a Test corpus for the known DataONE file formats.
- Define rules for DataONE file format Identification.
- Compile magic files for the Libmagic library used by the file command.

```
##------------------------------------------------------------------------
# EML (Ecological Metadata Language Format)
##------------------------------------------------------------------------
0 string <?xml
>&0     regex (eml)-[0-9].[0-9].[0-9]+ formatid="eml://ecoinformatics.org/%s"

##------------------------------------------------------------------------
# onedcx (DataONE Dublin Core Extended v1.0)
##------------------------------------------------------------------------
>&0     regex (onedcx/v)[0-9].[0-9]+ formatid="http://ns.dataone.org/metadata/schema/%s"


##------------------------------------------------------------------------
# FGDC-STD-001-1998 (Content Standard for Digital Geospatial Metadata, version 001-1998)
##------------------------------------------------------------------------
>&0     regex fgdc formatid="FGDC-STD-001-1998"

##------------------------------------------------------------------------
# Mercury (Oak Ridge National Lab Mercury Metadata version 1.0)
##------------------------------------------------------------------------
>&0     regex (mercury/terms/v)[0-9].[0-9] formatid="http://purl.org/ornl/schema/%s"

##------------------------------------------------------------------------
# ISOTC211 (Geographic MetaData (GMD) Extensible Markup Language)
##------------------------------------------------------------------------
>&0 regex  isotc211
>>&0 regex eng;USA formatid="http://www.isotc211.org/2005/gmd"
```

Apache Tika Detector & Parser

- Create custom-mimetypes.xml file for identification using Tika.
- It uses magic numbers as well for identification.
- Tika performs detection of the file type and based on that uses parsers for metadata extraction.
- Created custom namespace aware parsers for extraction of the metadata content from different file formats.
- Created a command line application based on Tika, which uses a configuration file for extracting standard set of metadata fields based on the file type.
- It takes file as input and identifies the file format and extract metadata properties.

```
--------------------------------------------------------------------------------
FILE FORMAT :text/xml; formatid="http://www.isotc211.org/2005/gmd"
--------------------------------------------------------------------------------
Title: Herring Infection Prevalence Data, 2007-2016, EVOS Herring Program
Creator: Paul Hershberger; Paul Hershberger; Paul Hershberger; Paul Hershberger
Subject: Clupea pallasii; Pacific herring, arenque del Pacífico; Prince William Sound; Exxon V
EVOSTC; EVOS Herring Survey; EVOS Herring; EVOS Herring Research and Monitoring; infection; vi
sis; disease prevalence; VEN; VHSV; Ichthyophonus; disease; Pacific herring; OCEAN > PACIFIC O
ORTH AMERICA > UNITED STATES OF AMERICA > ALASKA
Description: These data are part of the Herring Program of the Exxon Valdez Oil Spill Trustee
4120111-K, and 16120111-K, which is a multi-faceted study to determine why herring populations
1990s. The project was designed to assess the prevalence and intensity of several pathogens (v
c necrosis (VEN), and Ichthyophonus) in herring populations from 2012 to 2016 in Prince Willia
tka, Alaska were periodically sampled outside PWS. The dataset is ten tabular data files in co
ers including fish length, weight, positive / negative data for each pathogen, and pathogen lo
 2016) and sample location (Stika and PWS). Additionally, a single csv file contains a summary
us) for PWS from 2007 to 2016. Data from 2007 to 2011 were collected under the EVOS PWS Herrin
Publisher: Carol Janzen
Publisher: Alaska Ocean Observing System
Date: 20170101
Identifier: International DOI Foundation (IDF); 10.24431/axds/b3a5b45a-ae65-4d82-bca4-9636bed2
Coverage: -148.516027; -145.945612; 59.975405; 61.261070
Coverage: Time frame represents the bounding range for files named: 2012 PWS.csv, 2013 PWS.csv
, 2013 Sitka.csv, 2014 Sitka.csv, 2015 Sitka.csv, 2016Sitka.csv; 2012-01-01T00:00:00-09:00; 20
016.csv; 2007-01-01T00:00:00+00:00; 2016-01-01T00:00:00+00:00
```

References

We have created this template with scientific researchers in mind and with the help of feedback we have received. We encourage any comments or suggestions so that we can continue to update and improve this template. Visit this page to make a suggestion.

Acknowledgments

Check to make sure you've acknowledged partner and funding agencies, either with text or with their logos.

Introduction

- Identifying the correct file format is imperative for processing its contents.
- Many metadata standards are serialized as XML requires additional details of namespace information for processing.
- Packaging data into data packages requires metadata identification and parsing of the files.
- A tool for reliable identification makes it easier.

Aim:

- Determine the scientific resources using the Linux file command and Apache Tika which are excellent tools for file format identification.
- Use Apache Tika for parsing the metadata contents of the resources.
- Extraction of standard set of properties from the metadata.

File Command:

- File command performs several additional tests for determining the file format instead of using the file extensions.
- It uses the format signatures, known as magic numbers for identifying the file format.
- The magic directory contains the files, these files consist of the magic numbers. File command uses a compiled binary file containing the magic files.

Apache Tika:

- It is an open source toolkit for detecting and extracting metadata and contents of the files.
- Its ability to detect and parse file formats from over a 1000 different formats makes it a useful tool for search engine indexing, content analysis, translation etc.
- The new file types can be detected by creating a custom XML file containing the information.
- New parsers can be easily created and integrated into the application for fresh file formats.

Method

- Create Magic files for identifying DataONE file formats.
- Gathered a Test corpus for the known DataONE file formats.
- Define rules for DataONE file format Identification.
- Compile magic files for the Libmagic library used by the file command.
- Create custom-mimetypes.xml file for identification using Tika.
- It uses magic numbers as well for identification.
- Tika performs detection of the file type and based on that uses parsers for metadata extraction.
- Created custom namespace aware parsers for extraction of the metadata content from different file formats.
- Created a command line application based on Tika, which uses a configuration file for extracting standard set of metadata fields based on the file type.
- It takes file as input and identifies the file format and extract metadata properties.

Results

- Successful identification of the file types using Libmagic and Apache Tika.
- Used Python for unittest and the latest file version will contain the changes.
- Configurable command line application for detecting file types.
- Configurable tool for extracting desired set of metadata fields from the input file type.
- Configuration file helps in addition of new file formats and the respective metadata fields for extraction.