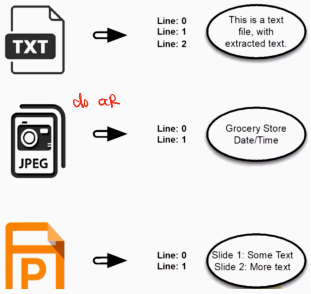
Content Extraction

Agenda

- How to extract text from any document?
- Basic Approach
  - Identification of Text
  - Analysis of Text
  - Featurization
- How is it done in Tika?
  - Architecture
  - Implementation and Tradeoffs
- Current directions in extraction of text

Text Extraction from File Types

- 1400+ file types
- Possibly different parsers for each type
- For each type, it has file header, body/ payload
- How to get text from each file type?

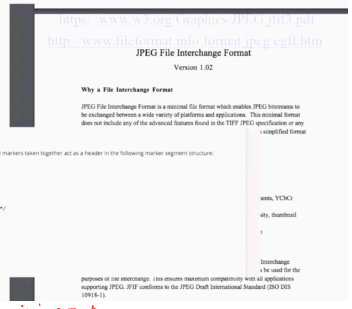


File Formats

- Text – Easy
  - Detect the language/charset and move to analysis
- Binary File Formats – Documents
  - Read File Specification
  - Identify Header, Content area
  - Extract text, move to analysis
- Binary File Formats – Multimedia and other content
  - Read File Specification
  - Identify Metadata
  - Possibly identify text in images/video and extract via image processing and analysis (e.g., Optical Character Recognition)

JPEG File Interchange Format

- JPEG FIF format
- Format specified as a file structure



Basic Approach

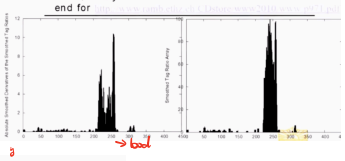
1. Identification of text
   - File Header – Usually Metadata information
   - Body/Payload of the file – File Specification can detail this – where is this in the file?
2. Analysis of Text
   - Structural Analysis – e.g., Tag Ratios (WWW2010), Emphasis (IRI2014)
   - Contextual Analysis
     - Proximity, stylistic (IRI2004)
3. Featurize
   - TFIDF (CACM 1975 – Salton), Clustering (Similarity), Metadata Extraction, Named Entity Recognition (NER)
   - BoilerPipe algorithm WSDM 2010

Tag/Annotation Ratios

- From WWW2010
  - Weninger et al.
- Take a web page, remove its scripts, stop words, styles, etc.
- Compute ratio of non annotated text (chars) with annotated (tags) chars
- Lines with more text than tags will have a higher ratio number
  - Can be used to isolate where text is in the HTML file, and to go after those lines

Algorithm 1 Compute Tag Ratios

```
INPUT: D
OUTPUT: T
D ← removeScriptTags(D)
D ← removeRemarkTags(D)
D ← removeStyleTags(D)
for all i ← 1 to |D| do
    x ← nonTagChars(D_i)
    y ← tags(D_i)
    if y = 0 then
        y ← 1
    end if
    T_i ← x/y
end for
```



Tag Ratios Algorithms in General

- Also called "histogram clustering"
- Build a text to tag ratio array (TTRArray)
- Use following procedure
  - Remove empty lines and script tags
  - Initialize the TTRArray
  - For each line in document
    - X = non tag ASCII characters
    - Y = # of tags in the line
    - TTRArray[current line] = X if no tags, else TTRArray[current line] = X / Y

Tag Ratio: Clustering

- Obtain article text by clustering
  - K-means
  - Expectation Maximization
  - Farthest First
  - Threshold clustering; using standard deviation as a cut-of threshold
- Higher the TTR is for an element relative to the mean TTR, more likely line represents a line of content-text

Featurize

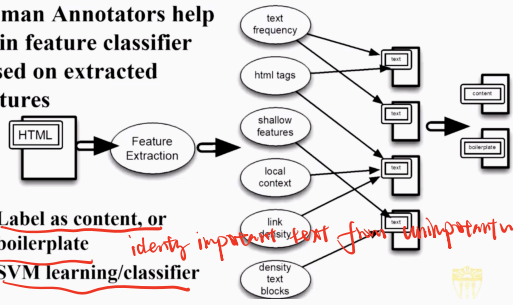
- BoilerPipe Algorithm – WSDM 2010
  - "Boilerplate Detection using Shallow Text Features" by Christian Kohlschütter et al., presented at WSDM 2010
- Train a classifier on both the text and structural features of a document to decide whether the observed piece of the document is "text" or "boilerplate"
- Two main ideas – look at *number of words* and *link density* as main features to determine what is text versus boilerplate

BoilerPipe – Main Features

- Text frequency of the whole corpus for common phrase extraction
- Presence of particular tags that enclose a block of text (heading tags, paragraphs, anchors, divs)
- Shallow features such as average word length, sentence length, absolute number of words in a defined segment
- Local context of the text – absolute/relative
- Heuristic features (number of words that start with uppercase, all caps, date and time tokens, link density)
- Density of text blocks

BoilerPipe Algorithm

- Human Annotators help train feature classifier based on extracted features



  - Label as content, or boilerplate
  - SVM learning/classifier

Readability – Browser Plugin

- https://web.archive.org/web/20110427053555/http://code.google.com/p/arc90labs-readability/
- Examines HTML and relies on article text, title, images, and well formed HTML to extract article text, even if segmented across multiple pages
- Ports to Python and Ruby, though originally developed in Javascript
- Arc90 experiment – NYC based design and technology shop started in 2009
- Authors: Philip Forget and Chris Schomaker
- http://www.readability.com/

Other Key Insight: Applications

- You need this: to read this:

Some Design Points

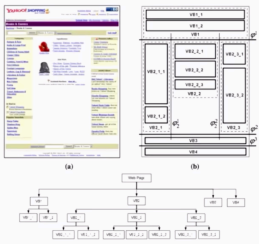
- Quantitative
  - Project Activity (num active committers; num commits)
  - Computational Performance
  - Memory Performance
  - Reliability
- Qualitative
  - Support for corner cases
  - Support for advanced text extraction
  - Metadata Support
  - License for the library
    - Permissive versus Reciprocal

Tika General Approach

- Curated mapping of IANA MIME types to parsers
  - Parser libraries that support MIME types are added to Tika, either in pure library form, or by writing small parser code
- Parser selection strategy
  - First parser selected that declares support for MIME type
  - FallBack Parser, CompositeParser, ExternalParser
  - Other methodologies are being explored
    - http://wiki.apache.org/tika/CompositeParserDiscussion

Visual Page Segmentation

- Cai et al, APWeb 2003
- Approach
  - Extract DOM blocks and assign them a value based on visual coherence
  - Build hierarchical model of the site based on this
  - Vertical/Horizontal separators
  - Identify content blocks



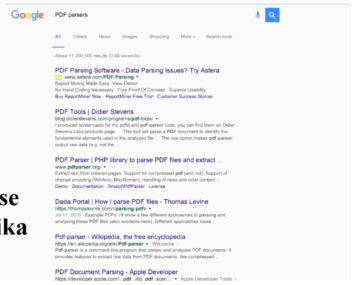
https://web.archive.org/web/20110612200136/http://www.cmucdeg.cs/deng/cai-APWeb03.pdf

Some Key Insights and Observations from the Research

- Several tried and true approaches work when you can deal with HTML oriented documents
  - Is there some way to transform documents into HTML, especially well formed HTML?
- Lots of exploitation of text oriented features, and text summarization techniques
  - Metadata features not really used, since focus in on content – we'll get to metadata later
- Many libraries exist to perform extraction
  - Some are more powerful than others, more reliable, etc.

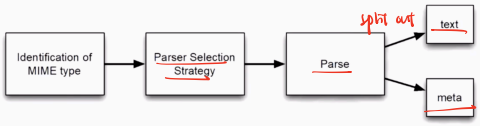
Finding Parsing Libraries is Difficult

- Google search for PDF parsers
- 50+ parsers on the first few pages
- Which one is the right one?
- Note: NONE of these are the ones that Tika uses (PDFBox)



Apache Tika – Content Extraction Process

- Looks something like this



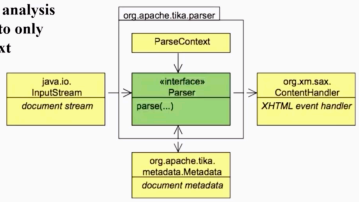
- Incorporates Parser Selection based on MIME hierarchy
- Requires mapping of Parsers to MIME types that they support

Design Goal: Convention over Configuration

- AutoDetectParser algorithm
  - Load all parsers
    - For any parser that fails to initialize or where JAR file and/or class isn't available, ignore or warn and move on
  - Load all mappings to MIME types
  - Incoming file, stream, URL
    - Detect MIME Type with Accuracy
    - Find first parser that satisfies MIME type, and invoke it
    - Parser could be Façade to
      - Fallback Parser (try one; fall back to other); Composite Parser (call many parsers in some sequence); External Parser (external program)

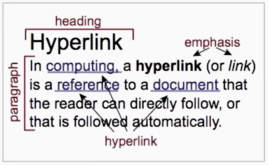
XHTML as an intermediate output format

- Wealth of research on how to process text from HTML documents
- XHTML => structured HTML, well formed unlike most of Internet
- Can leverage streaming SAX (Simple API for XHTML Processing) architecture
- Enables wide text analysis and isolates Tika to only extracting base text



Another benefit

- XHTML allows annotations to be captured surrounding text
  - Annotations are key input e.g., to Tag Frequency algorithms, to Feature Detection Algorithms and to Text Analytics
  - Also extremely useful from a metadata perspective as annotations can enhance metadata extraction



Some Current Thrusts and Research Areas

Optical Character Recognition

- Tesseract
  - https://code.google.com/p/tesseract-ocr/
  - Great and Accurate Toolkit, Apache License, version 2 ("ALv2")
  - Many recent improvements by Google and Support for Multiple Languages
- Integrated Tika and Tesseract
  - http://issues.apache.org/jira/browse/TIKA-93
  - Thank you to Grant Ingersoll (original patch) and Tyler Palsulich for taking the work the rest of the way to get it contributed

Tika + Tesseract

- https://cwiki.apache.org/confluence/display/tika/TikaOCR
- brew install tesseract --all-languages
- tika -t /path/to/tiff/file.tiff
  - Tika will automatically discern whether you have Tesseract installed or not
- Try it from the Tika REST server!
  - In another window, start Tika server
    - java -jar /path/to/tika-server-1.11.jar
  - In another window, issue a cURL request
    - curl -T /path/to/tiff/image.tiff http://localhost:9998/tika --header "Content-type: image/tiff"

Scientific Literature

- GeneRation Of Bibliographic Data (GROBID)
  - P. Lopez et al, Research and Advanced Technology for Digital Libraries, 2009
- Uses Conditional Random Fields and trained on 1000 examples for identifying header information (authors, institutions, publication year, affiliations) and bibliographic information (citations) and measurements from scientific literature
  - GROBID code on Github:
  - http://grobid.readthedocs.org/en/latest/Introduction/

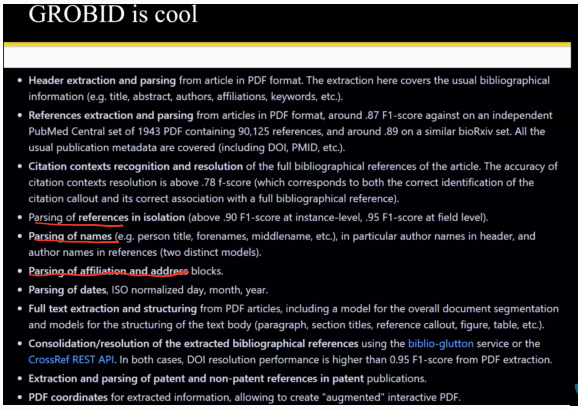
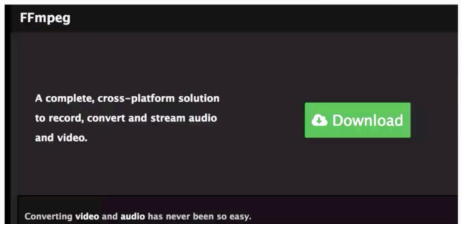


Image / Video / Multimedia

- Extraction of content using traditional image and video analysis tools
  - FFMPEG, EXIF Tool



Scientific Analysis

- GeoTopicParsing
  - Text Analysis with Apache OpenNLP to identify Location entities in text
  - Lucene Geo Gazetteer
    - Entities resolved to Geonames and to Latitutdes and Longitudes
    - https://github.com/chrismattmann/lucene-geo-gazetteer
  - Result is: "reading" the text to spot locations and to return their coordinates
  - Started as a CSCI 572 Directed Research Project!
- GDAL (Geospatial Data Abstraction Library)

Tika Research in Advanced Extraction

- https://cwiki.apache.org/confluence/display/tika/#Home-AdvancedContentExtractionwithTika-Integration

Advanced Content Extraction with Tika - Integration

- Getting Tika up and Running with Pooled Time Series - How to use Tika with the Pooled Time Series video descriptor similarity code.
- Getting Tika up and Running with Apache cTAKES - How to use Tika with Apache cTAKES the clinical text biomedical knowledge extraction framework.
- Getting Tika up and Running with EXIFTool - How to use Tika with EXIFTool.
- Getting Tika up and Running with FFMPEG - How to use Tika with FFMPEG.
- Getting Tika up and Running with the GROBID PDF Journal parser - How to use Tika with the GROBID PDF journal parser.
- Getting Tika up and Running with the GeoTopicParser based on Geonames.org, Lucene, and OpenNLP
- Getting Tika up and Running with OCR - How to use Tika with OCR from Tesseract.
- Getting Tika up and Running with the Geospatial Data Abstraction Library (GDAL) - How to use Tika with GDAL to parse/extract geospatial data files.
- Getting Tika up and running with Stanford Core NLP and with OpenNLP - How to use Tika with Stanford NER/NLP and with Apache Open NLP.

Summary

- How to extract text from any document?
- Basic Approach
  - Identification of Text
  - Analysis of Text
  - Featurization
- How is it done in Tika?
  - Architecture
  - Implementation and Tradeoffs
- Current directions in extraction of text

API Approaches for Text Extraction

- Alchemy
  - http://www.alchemyapi.com/api/text-extraction
    - HTML to XML, JSON and RDF
- Textise
  - Any page or URL to text
  - http://www.textise.net/
- Givemetext.org Open Knowledge Foundation
  - http://givemetext.okfnlabs.org/
  - Based on Apache Tika