Senior Project, 2018, Fall

Web Page Archiving and Content Analysis

Student: Oscar Martinez, Florida International University
Mentor: Mark Finlayson, Florida International University
Professor: Masoud Sadjadi, Florida International University

FIU School of Computing & Information Sciences

MIT LINCOLN LABORATORY

Problem

- Temporary web content
- Filtering and parsing is time consuming and difficult
- Content may need translation
- Content not available for offline access and manipulation
- No solution exists

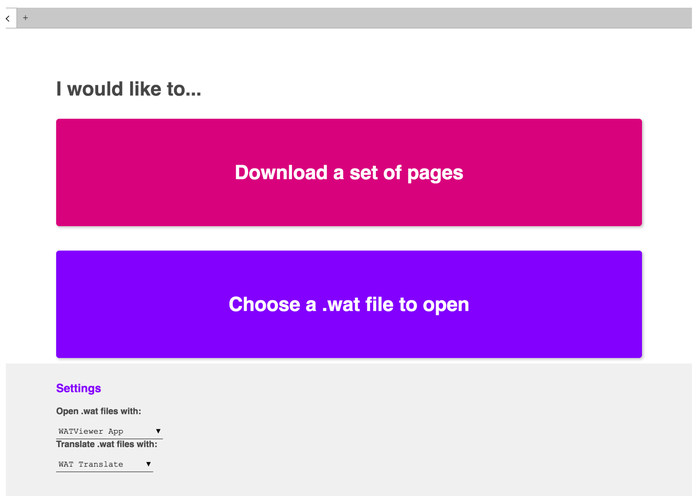
Solution

- Custom file format
- Improved GUI for viewing and interfacing with custom file format
- Introduced deep learning translation API
- Introduced machine learning approach to content extraction
- Cross-platform

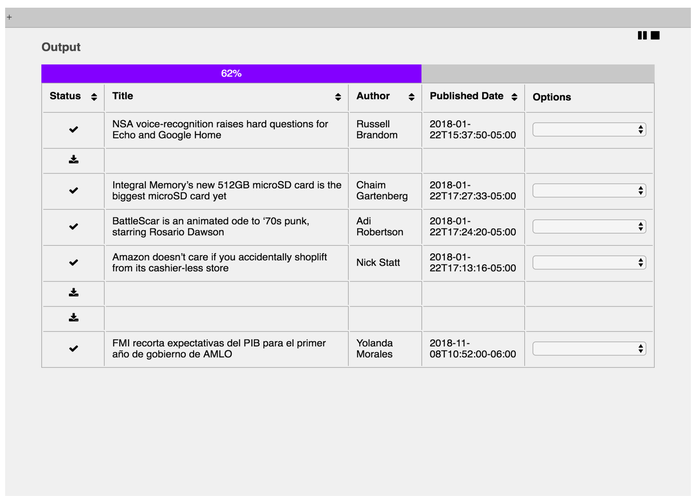
Implementation

- Python2.7 and wget for the downloading CLI
- Electron, HTML5, CSS3, and JS for the GUI
- Dragnet for the content extraction
- PyTorch and OpenNMT for the translation

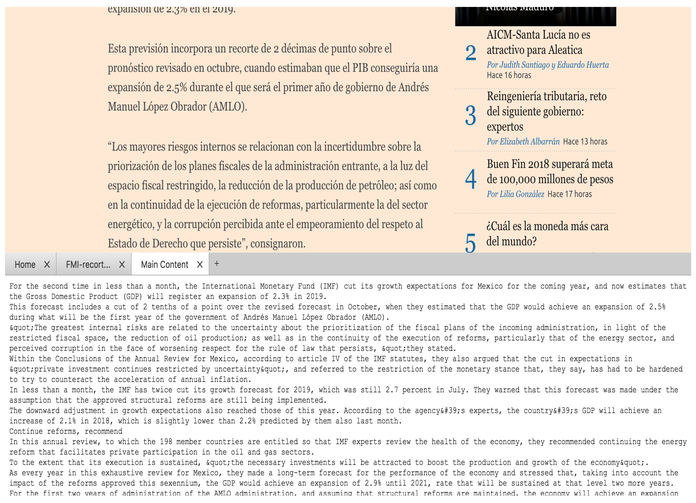
Screenshots



Main Screen



Downloading Screen



Translated/Extracted Content

Current System

- Previous projects were incomplete
- Existing partial solutions have been deprecated

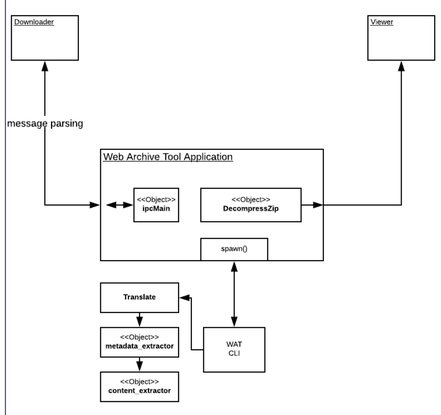
Requirements

- Batch download large sets of pages
- Encapsulate sets of pages in a session
- Continue session if download was interrupted
- Archive web page in format that allows easy programmatic access
- Encapsulate this archive in a single file
- Ability to easily view web archive
- Identify, translate, and extract the main textual content of the article
- Identify and extract the metadata of the article
- Cross-platform

Verification

- Unit and integration testing
- Tested on 5GB of articles
- Tested on macOS, Windows, and Linux
- Content Extraction: 90% accuracy
- In-house Translation: 70% accuracy

System Design

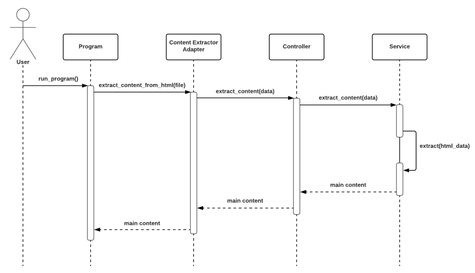


Overview of System

Content Extraction Algorithm

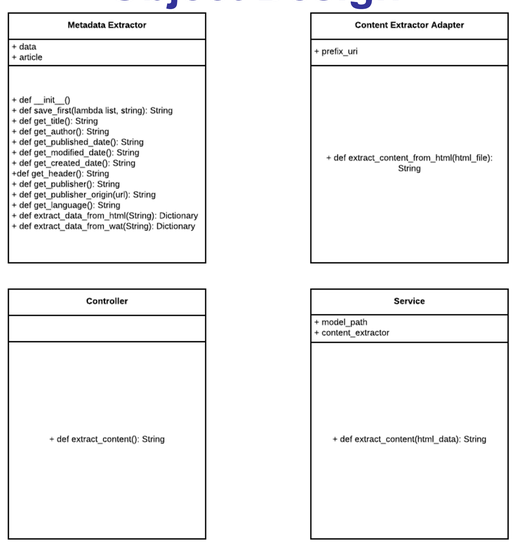
Dragnet Content Extraction Algorithm

1. *Split each web page into a sequence of blocks by traversing the DOM*
   a. *Create a new block whenever an HTML tag (e.g. div, p, h1, etc) is encountered*
2. *For each block, construct a set of features to be used in a classifier to predict and extra block level content/no-content*
   a. ***Text and link density**: content blocks generally have more text density and less link density than non-content blocks*
   b. ***Attribute names**: programmers will leave behind useful semantic information about a block's content by using descriptive names for the attributes*
   c. ***Content-tag ratio**: ratio of text length and HTML tags tends to be higher in content blocks while non-content blocks have lower content-tag ratios. Using unsupervised k-means clustering, blocks are clustered based on their content-ta ratio*
3. *Return main content*



Overview of Content Extraction

Object Design



Summary

Our work allows users to:

- Download and archive batches of web pages
- Translate and extract important information from the web pages
- View these web pages on any platform

Acknowledgment

The material presented in this poster is based upon the work supported by Mark Finlayson, Andres Cremisini, and Pedro Torres-Carrasquillo. I am thankful for the help that I received from my group member, Adam Tahoun.

Follow us @FIUSCIS