Checkpoint 1 - Team UFO

An Analysis of Declassified Government Documents

Cory Maclauchlin, John Myers, Arivirku Thirugnanam

6/3/2022

Introduction

There has been a recent uptick in the conversations surrounding the notion of "Unidentified Flying Objects," or UFOs, in the media due to Congressional inquiries on the topic. The United States Government Freedom of Information Act (FOIA) allows individual citizens the right to ask for and receive previously unreleased documents possessed by the Government upon request. When these documents are released according to the law, "Internet detectives" pour through the documents to find that hidden nugget of information. In the case of this project, we want to see if there are any hidden patterns surrounding the origin of these so-called UFOs that the Government hasn't disclosed before. We will apply our developed system to a series of documents released by the Central Intelligence Agency (CIA) through a FOIA request to better understand the nature of the data.

Problem Description

Some of the technical challenges of this project include:

- **Large Troves of Documents.** When released, a trove of documents is published all at once. These can include hundreds of PDFs with thousands of pages. For this particular release of documents from the CIA, there are 712 documents composed of 3493 total pages.
- **Data That is Dirty.** The Government releases documents that are scans of printed materials, with redactions made. This is to ensure that nothing is accidentally revealed that isn't supposed to be, so a physical step is required in Government release procedures. As such, any useful digital representation that may have existed is obliterated. The scans tend not to be sophisticated or high-quality, either.
- **Unusual Lexicon.** A standard dictionary methodology may not be appropriate for certain document types, especially this dataset. The unusual lexicon surrounding both alien and terrestrial technology will require an unsupervised approach to performing processing.
- **No Categorization or Labeling.** Documents are released without any labeling or categorization of any kind. This leaves it as an exercise for the recipient to pour through the mounds of information to find the needle in the haystack they are looking for.

Software Development

Our development methodology will be a standard functionally decomposed web application, with both backend and frontend development being accomplished. The goal is to reduce coupling between elements of our system and increase cohesion on the responsibilities of the designed subsystem. We have performed an initial survey of the requirements of the system, along with a design and implementation approach, which will now be discussed in detail.

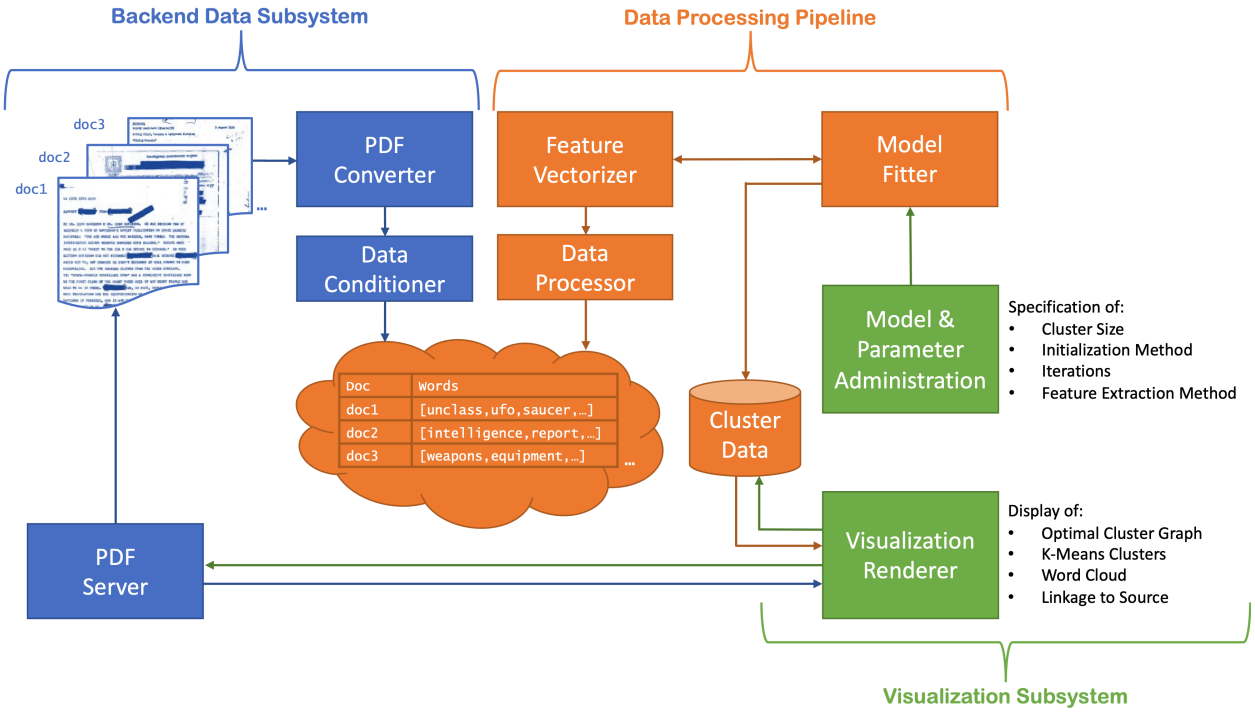
Requirements

This tool will be designed generically to accommodate any type of cache of FOIA-released documents from the Government in PDF form. Entire directories of documents will be processed within.

There will be two primary user classes for the developed software. The first user class is the *data scientist* who will set up the backend for data processing, including administering the parameters for the model generation steps. The second user class is a *citizen data consumer* interested in understanding the document set, especially for finding the hidden patterns within the data. In addition, this user class will leverage the clusters that have been exposed.

Design

The overall architecture will be designed around three subsystems and pipelines, the Backend Data Subsystem (in Blue), the Data Processing Pipeline (in Orange), and the Visualization Subsystem (in Green):



Backend Data Subsystem

The backed data subsystem will be responsible for opening all of the OCR'd PDF files within the specified directory, cleaning them as appropriate, and building the interim data format for the data processing pipeline.

1. Documents are parsed and imported into memory.
2. All words will be compared to a Wordnet, and only words with semantics will be indexed.
3. Only documents that ultimately have more than two actual words will be indexed.

The goal of these conditioning steps is to remove the numerous incorrectly recognized words in some of the more poorly scanned documents. This will result in a more accurate representation of the hidden patterns within the documents. Finally, as a stretch goal, we will have an accessor available so that a user can choose a specific document from a cluster displayed on the rendered visualizations and see the contents of the original (source) document that was used.

Data Processing Pipeline

The data processing pipeline will be responsible for building all elements required for indexing, querying, and building clusters of our data.

1. Build features of the data set by leveraging a count of the words used in the documents.
2. Calculate the Sum of Squared distances for a range of clusters, likely from 2 to 20.
3. Determine the optimal number of clusters through analysis of the "elbow method."
4. Cluster PDFs by fitting the model to the entire data set.

This pipeline will allow the modification of a number of parameters around this clustering technique. A few examples of these include the initial cluster space, often done via random seeding. A second is the number of clusters that the documents will be fit to. Finally, the word frequency for a given cluster as presented to the user. In addition, this data processing pipeline will create all appropriate indexes so that querying can be accomplished through the web-based frontend.

Visualization Subsystem

The visualization subsystem will be responsible for displaying user interfaces that will allow for both the administering of the data processing pipeline parameters and also the visualization of the results of the analysis. Modifiable parameters in the User Interface will include:

- Cluster Size
- Clustering Initialization Method
- Number of Iterations
- Feature Extraction Method

Once parameters have been changed and processing is complete, the following representations will be displayed to the user as a result:

- Optimal Cluster Graph
- Applied/Computed Clusters
- Word Cloud
- Linkage to Source PDF

Implementation

We intend to heavily leverage existing libraries to supplement our implementation of this project. Some initial libraries that we've identified include:

- Apache PDFBox - Open and process text data within an OCR'd PDF document.
- Python & Scikit Learn - Data Processing Pipeline and rendering of glyphs.
- reactJS or python Django - Web Frontend for Visualization Subsystem.

Test & Demonstration

We will use the 712 PDFs that have been published here: https://documents2.theblackvault.com/documents/cia/CIAUFOCD-FULL-CONVERTED.zip. These documents have been made searchable from the original release by the CIA under a FOIA request. In addition, several other document sets have been published in more recent days that may be an exciting check of the generic design of our system. These include a cache released from the Navy here: https://www.secnav.navy.mil/foia/readingroom/CaseFiles/UFO%20Info/UAP%20DOCUMENTS.