SAS6740-2016

Text Mining Secretary Clinton's Emails

Michael Ames, SAS Institute Inc., Cary, NC

ABSTRACT

The recent controversy regarding former Secretary Hillary Clinton's use of a non-government, privately maintained email server provides a great opportunity to analyze real-world data using a variety of analytic techniques. This email corpus is interesting because of the challenges acquiring and preparing the data for analysis as well as the variety of analysis that can be performed, including techniques for searching, entity extraction and resolution, natural language processing for topic generation, and social network analysis. Given the potential for politically charged discussion, rest assured there will be no discussion of politics—just fact-based analysis.

INTRODUCTION

On March 2, 2015, *The New York Times* (Schmidt 2015) broke the story of how Secretary Clinton had exclusively used a private email server rather than a government-issued one throughout her time as Secretary of State, and that her aides took no action to preserve emails sent or received from her personal accounts as required by law. This has led to an ongoing political controversy the likes of which haven't been scene in US politics since the "Penny Debate" of 1990 (HR 3761-1989). While the politics of the controversy are interesting, they aren't the purpose of this paper. Instead, this paper deals with analyzing publicly available data and unstructured text.

This paper discusses the following topics:

- acquiring, preparing, and converting the "Clinton Emails" on the State Departments FOIA site
- cleansing, analyzing, and descriptive statistics
- text mining, entity extraction, and topic generation

While the subject of this analysis, the Clinton Emails, can be seen as controversial, it highlights the challenges of dealing with real-world data and, more importantly, "open" publicly available data. While email is the subject of this analysis, the tools and techniques used to prepare and analyze these emails can be applied to any unstructured data. One of the key areas of analysis, Entity Extraction, is an often overlooked feature of text mining. This set of documents contains a rich set of People, Places, and Organizations to extract and analyze.

ACQUIRING EMAILS FROM THE FOIA READING ROOM

The first step in the analysis of the emails to acquire them from the "Clinton Collection" at the State Department's Freedom of Information Act (FOIA) reading room, available at https://foia.state.gov. Interestingly, they don't make it easy to just download all the emails in one package. Instead, each month since May of 2015, the government makes a number of declassified and redacted emails available. As of March 2016, 30,322 emails had been released.

To acquire the emails from the FOIA reading room, a Python program was used to first get the list of available emails, download the PDF images of the emails, convert the PDF file to text, and finally insert it into a database from which the analysis takes place.

For the initial analysis, a PostgreSQL table is created using Python to collect the raw data:

```
crTbl = """
    CREATE TABLE clinton_email (
        email_id integer primary key,
        email_pdf varchar,
        email_from varchar,
```

```
            email_to varchar,
            email_date date,
            pdf_url varchar,
            email_subject varchar,
            case_no varchar,
            email_content text
            )
    """
```

Next, the request is made returning a JSON result also with Python:

```
response =
requests.get("https://foia.state.gov/searchapp/Search/SubmitSimpleQuery",
    params = {"searchText": "*",
              "beginDate": "false",
              "endDate": "false",
              "collectionMatch": "Clinton_Email",
              "postedBeginDate": "false",
              "postedEndDate": "false",
              "caseNumber": "false",
              "page": 1,
              "start": 0,
              "limit": 100000},
    verify=False)
```

Now that the target table has been created, and the results from the query has been returned, a loop is set up to cycle through the result set and insert it into the PostgreSQL table:

```
for row in data["Results"]:
    # date handling for postgres
    forDate = fixDate(pStr(str(row['docDate'])))
    pdfFile = row['pdfLink'][-13:]
    pdfURL = "https://foia.state.gov/searchapp/" + row['pdfLink']
    cur.execute(insTable,(id, pdfFile, row['from'], row['to'],
                  forDate, pdfURL, row['subject'],row['caseNumber'], ''))
```

CONVERTING EMAIL PDF FILES TO TEXT

Once a list of all the emails has been inserted into the PostgreSQL table, the actual PDF files need to be downloaded, converted to text, and added to the base table. Python provides a number of PDF-to-text packages. However, many of them, like much in the open-source world, are not maintained or just plain don't work. The Python PDFMiner package seems to have a relatively active group of maintainers, so it was chosen to perform the conversion process. The Apache Tika package also seems to work well but had some key limitations in converting these PDF files.

A simple SQL query is used to get a list of PDF files to pull down with Python:

```
getList = """
    SELECT email_pdf, pdf_url, email_id
    FROM clinton_email_apr """
```

A function is used to download the PDF and write it to the file system with Python:

```
def getPDF(doc, url):
        pdf= urllib2.urlopen(url, context=ctx)
        output = open(doc,'wb')
        output.write(pdf.read())
        output.close()
```

A function is used to convert the PDF documents to text using Python PDFMiner library:

```
def convertPDF(fname, getStat, pages=None):
      pagenums = set(pages)
      output = StringIO()
      manager = PDFResourceManager()
      converter = TextConverter(manager, output, laparams=LAParams())
      interpreter = PDFPageInterpreter(manager, converter)
      infile = file(fname, 'rb')
      for page in PDFPage.get_pages(infile, pagenums):
            interpreter.process_page(page)
            infile.close()
            converter.close()
            text = output.getvalue()
            output.close
     return text
```

Finally, a FOR loop was used to tie the process together. For each row (email), get the PDF from foia.state.gov, convert it to text, and update the email table with the text that it contains:

```
rows = cur.fetchall()
for row in rows:
      pdf     = getPDF(row[0], row[1])
      pdfText = convertPDF(row[0], pdf)
      cur.execute(updtSQL, (pdfText, row[0], row[2]))
      con.commit()
```

CLEANSING, ANALYZING, AND USING DESCRIPTIVE STATISTICS

Now that the emails are safely downloaded to a PostgreSQL table, they can be analyzed with SAS®. First we'll want to clean up the senders and receivers as well as address any of the text conversion issues. On the analysis front, SAS® Text Miner and SAS® Enterprise Miner are used to extract entities of interest and generate topics. The HPENG procedure is used to "resolve entities," and using some of the graph capabilities, we can visualize some networks.

CLEANING UP SENDERS AND RECEIVERS

A quick analysis of the email senders and receivers shows significant data quality issues that we can easily address with the SQL procedure and/or the DATA Step. Here is an example breakdown of the email_to field for emails sent to Secretary Clinton that contain close to two dozen different aliases and at least 4 email addresses.

email_to	COUNT_of_email_id
H	17296
Hillary	196
H2	37
Secretary	31
HRC	26
Clinton, Hillary	17

Table 1. Sample of Email_to Field

To clean up the data, a table of aliases was created containing the “alias” to resolved name. Two simple PROC SQL statements are used to clean up the **email from** and **email to** fields.

```
proc sql;

     create table _email_senders
     as
     select a.*, b.person_name as email_from_name
from mydb.clinton_email_apr a left join mydb.clinton_alias b on (a.email_from = b.alias_name);

     create table _email_recievers
     as
     select a.*, b.person_name as email_to_name
from _email_senders a left join mydb.clinton_alias b on (a.email_to = b.alias_name);


quit;
```

ANALYZING TOP 10 SENDERS AND RECIPIENTS

With relatively clean sender’s and receiver’s names, visualizing is relatively straightforward with SAS® Enterprise Guide. The following tile graphs present the top 10 senders and recipients of emails in the document collection.

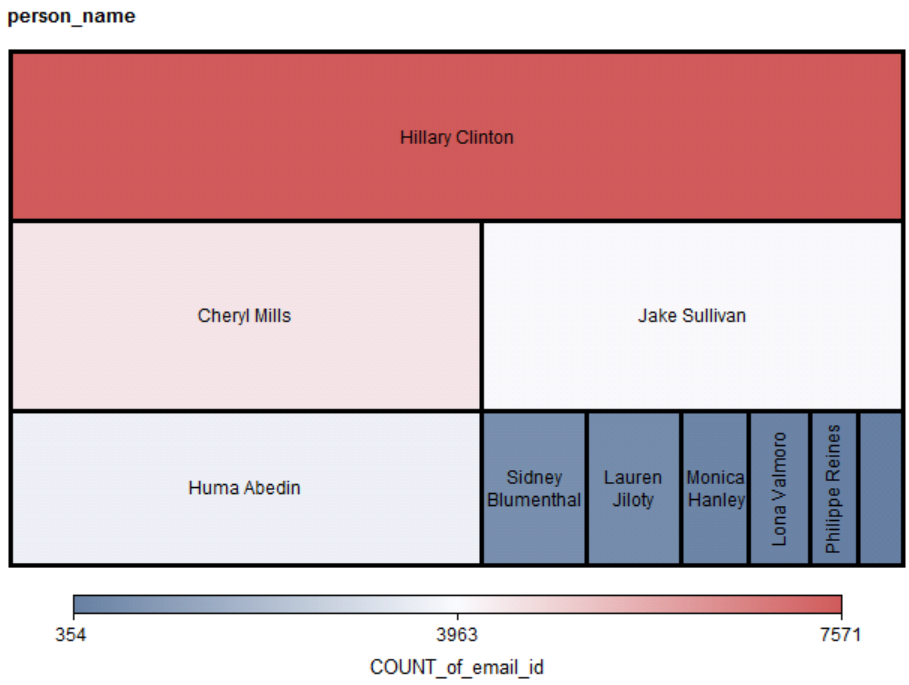

Figure 1. Top 10 Email Senders in the Document Collection

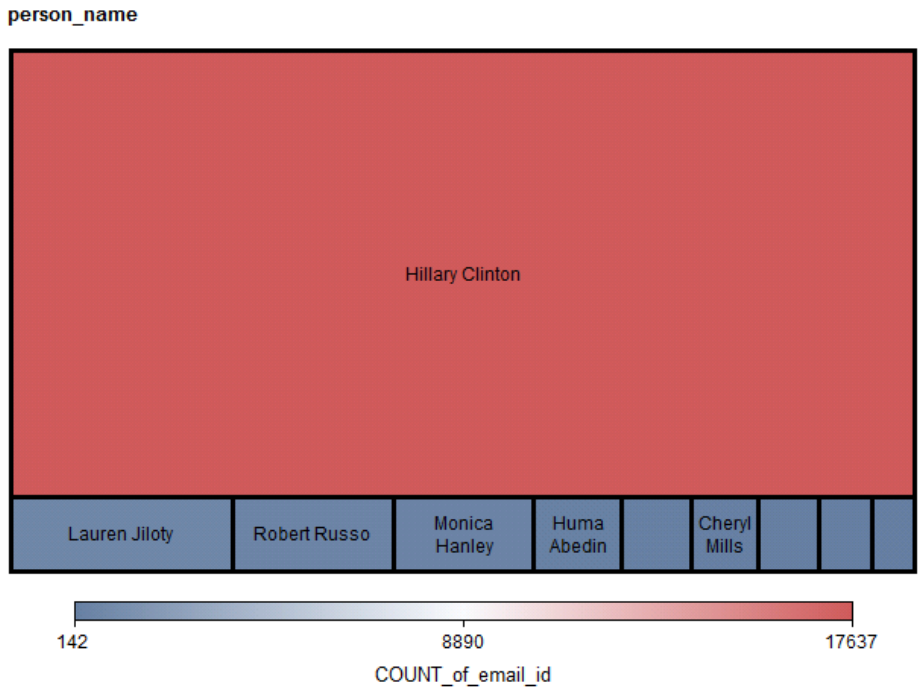

Figure 2. Top 10 Email Recipients in the Document Collection

ANALYZING TOP 25 SENDERS AND RECIEVERS

Analysis of Senders to Receivers is also interesting, as it shows most of the emails didn't originate from Secretary Clinton. Instead, she was the main recipient of emails.

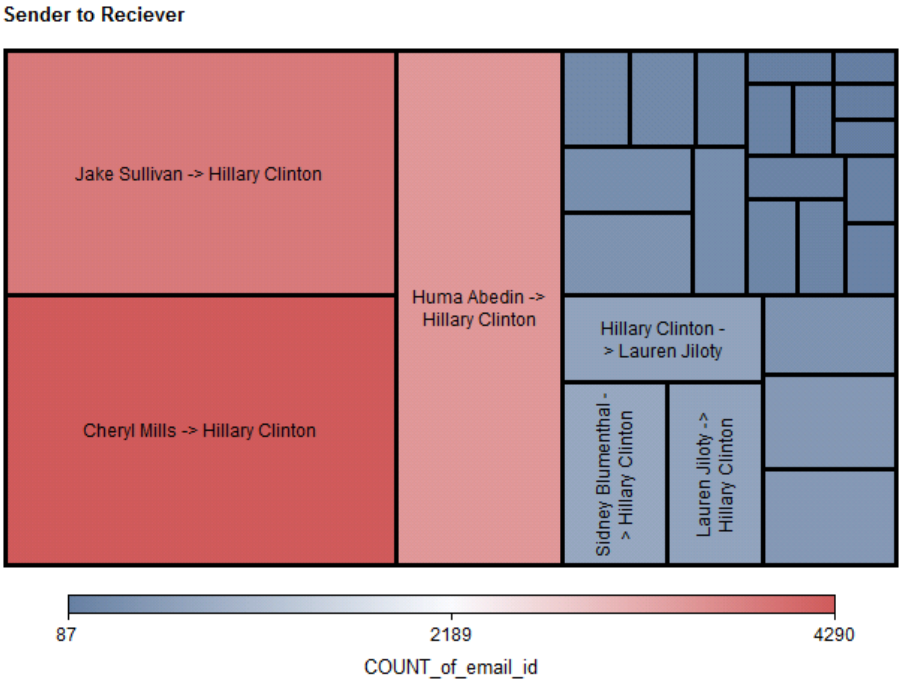

Figure 3. Top 10 Email Recipients in the Document Collection

ENTITY EXTRACTION WITH THE HPTIMINE PROCEDURE

EXTRACTING ENTITIES

One of the most overlooked features of SAS text mining capabilities is entity extraction. The interesting things inside the emails are people, places, and organizations mentioned. One of the really neat things to do with SAS text mining procedures is to produce tables of “People”, “Places”, and “Organizations” mentioned in the emails. The following HPTMINE procedure and Macro call produce an output table that contains the terms by document mapping “OUTTERMS”.

```
proc hptmine data=emails;
var email_content;
doc_id email_id;
parse
      outterms=outterms
      termwgt=entropy
      entities=std
  outparent=outparent;
svd
max_k=50
outtopics=topics
SVDU=u_matrix
TOPICLABELSIZE=7;
run;
```

Once the terms are rolled up by Entity Types Person, Location, and Organization, they can be analyzed.

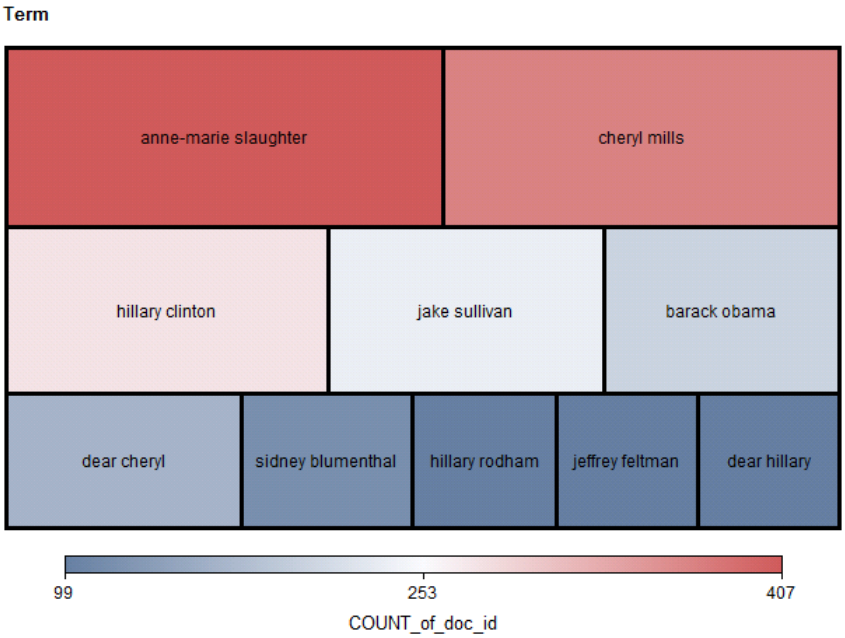

Figure 4. Top 10 People Extracted from the Document Collection

It is interesting that Jeffrey Feltman shows up as one of the top people mentioned in the documents but doesn't appear in the top sender or receiver list. Also, you can find different data quality issues such as "Dear Cheryl" or "Dear Hillary", which could have been resolved by using a custom entity extraction corpus.

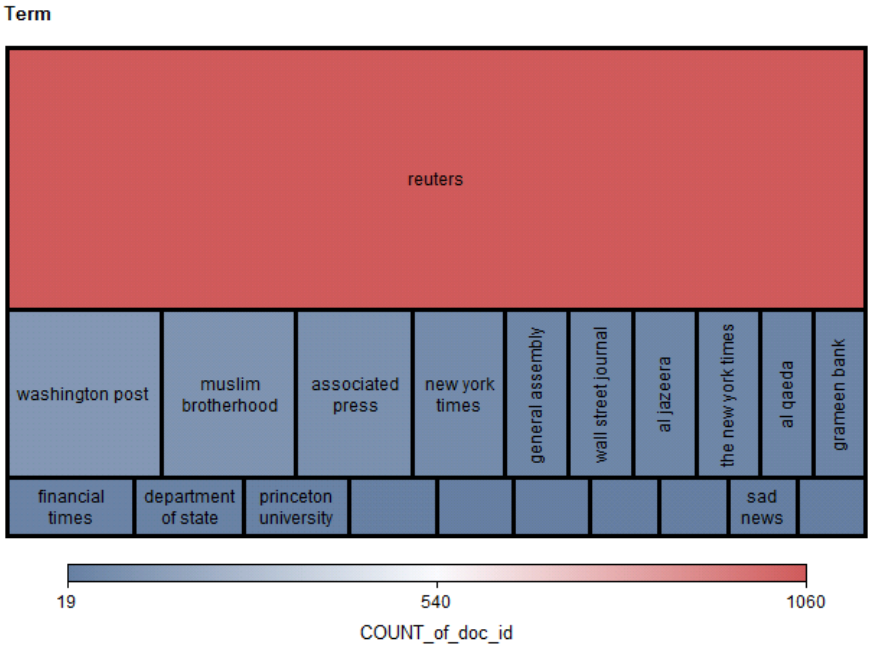

Figure 5. Top 20 Organizations Mentioned in Emails

The tile graph shows a breakdown of organizations mentioned. Interestingly, four out of the five organizations mentioned are news organizations.

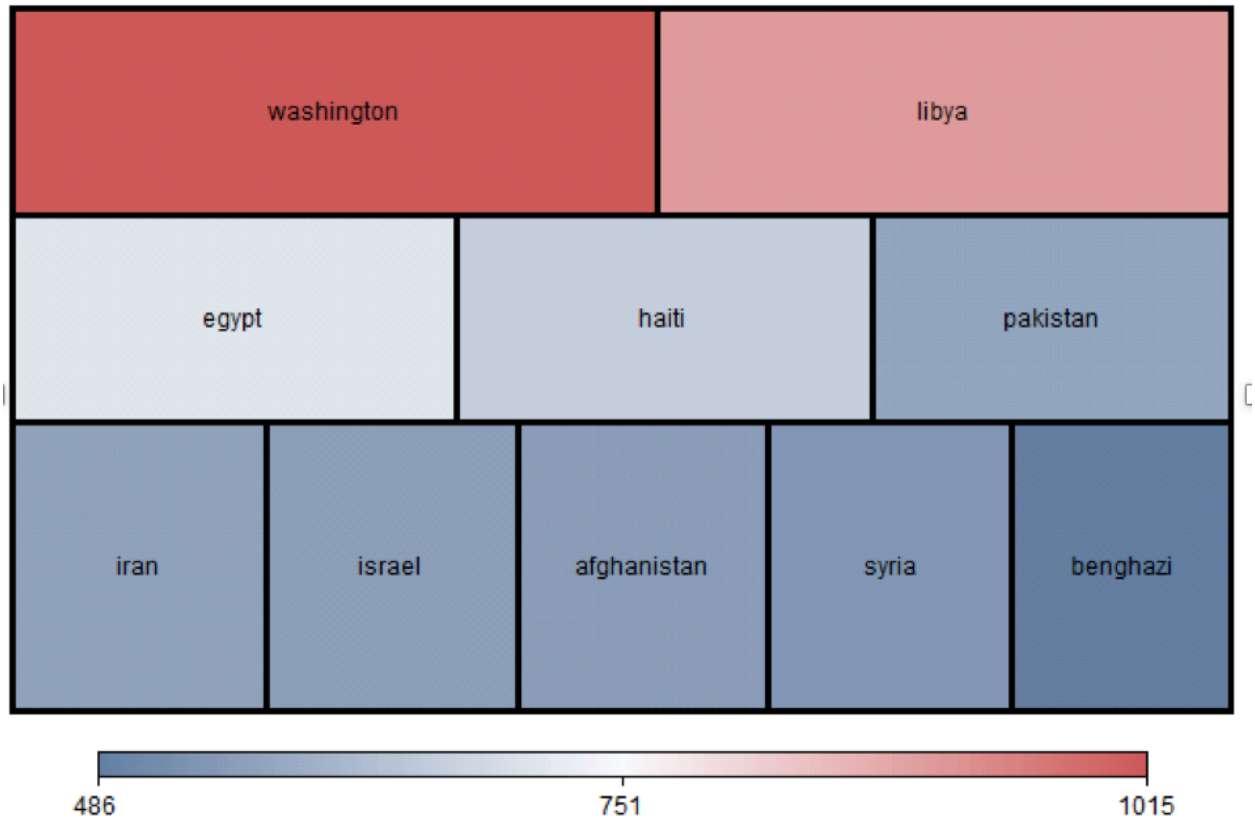

Figure 6. Top 10 Locations Mentioned in Emails

CONCLUSION

Dealing with real-world "text" data has challenges above and beyond what is found in so called "unstructured" data. Using a mix of SAS and Python code is becoming more common as both languages naturally complement one another. Expectations about the quality of open data and publicly available data quality have to be kept into mind as often the case, data quality of publiclly available data is questionable. Entity extraction is an often overlooked capability of SAS® Text Miner but a super useful feature that can be used to extend analysis beyond simple text parsing and word clouds

REFERENCES

H.R.3761 – Price Rounding Act of 1989 (Introduced in House – IH)

Schmidt, Michael S. "Hillary Clinton Asks State Department to Vet Emails for Release." *The New York Times*, March 5, 2015. Available http://www.nytimes.com/2015/03/06/us/politics/hillary-clinton-asks-state-dept-to-review-emails-for-public-release.html?ref=politics&_r=2.

ACKNOWLEDGMENTS

Special thanks to Sam Penfield, Stephen Siegert, and Gordon Robinson of SAS for help with analyzing the data and developing the presentation.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Michael Ames

SAS Institute Inc.
Michael.Ames@sas.com
https://www.linkedin.com/in/amichaelames