# ENRON EMAILS: PROCESSING TEXT FILES TO CREATE A USEFUL ANALYTICAL TOOL

Cynthia Vint

BU MET CS 779 Spring 1 2017

## Table of Contents

## Introduction

While Data Science and computational methodologies to extract actionable insights from various types of data are advancing at a rapid pace, ETL and data preprocessing in general still take up a huge amount of the average Data Science team's time. This is because data comes in so many different forms, and a massive proportion of data is either semi-structured or unstructured. Data management and quality has become such a huge task for most companies, that Chief Data Officers (CDOs) have become commonplace at the executive level[1].

This project delved into a specific type of data that comes in notoriously unstructured formats: text data. For obvious reasons, text data is complicated to clean and process at every step in the process of any Text Analytics job. Issues such as encoding, such as in multilingual text data, formatting, and extraction of important metadata can sometimes seem all but impossible. It always requires creativity, and is rarely a smooth process.

The data set chosen for this endeavor was the Enron Email dataset[2]. This data was released by the Federal Energy Regulatory Commission during its investigation of Enron in 2001 following their file for bankruptcy[3]. Its integrity was later improved by a team at MIT, and it is now publicly available on multiple websites thanks to their efforts[4]. This dataset is ideal for a study in processing text data sets because, firstly and most relevant, it is primarily text data.

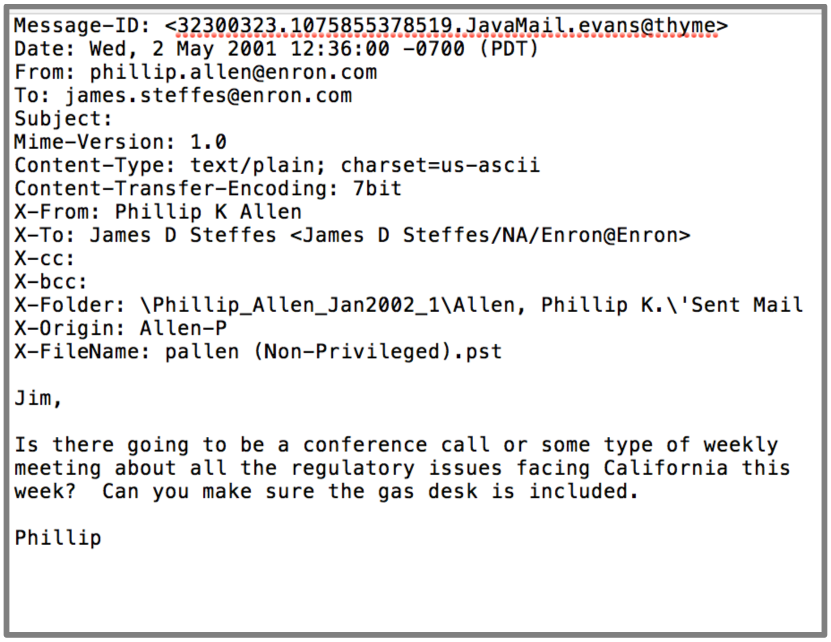
[1] Berkooz, How Chief Officers Can Get Their Companies to Collect Clean Data, https://hbr.org/2017/02/how-chief-data-officers-can-get-their-companies-to-collect-clean-data
2 Enron Email Dataset, https://www.cs.cmu.edu/~./enron/
3 Johnston, Enron's Collapse: The Investigation, http://www.nytimes.com/2002/01/16/business/enron-s-collapse-investigation-justice-dept-s-inquiry-into-enron-beginning-take.html
4 https://www.cs.cmu.edu/~./enron/

Secondly, it captures human networks and behavior in such a way that only a repository of communications will, rather than a collection of news articles or books. Finally, its size is ideal for at least scratching the surface in questions of scalability while still working within the limited means of a local machine, as many data scientists and analysts often must.

The results of this experiment provided a reusable approach to work with large amounts of text files and ultimately organize them into a query-capable format that can be used for multi-faceted analyses, both simple and advanced. The technologies used revealed their advantages and limitations, and the final product was utilized to reproduce, or at least paraphrase, if you will, some research that has been done on this dataset to date. It showed that such analyses can be performed quickly and easily by querying the simple database schema and reading the results into data analytical programs.

## The Original Data

The Enron dataset consisted of a 3-layered file system. At the first level were the usernames of the 150 or so users from whom the emails were taken. The second layer showed the folder organization of their email documents. Aside from ‘Sent’ and ‘Received’, there were also often personalized folders, draft and deleted folders, etc. Inside these folders were text files. These text files contained not only the emails’ text, but also the metadata. An example of a text file can be seen below:



```
Message-ID: <32300323.1075855378519.JavaMail.evans@thyme>
Date: Wed, 2 May 2001 12:36:00 -0700 (PDT)
From: phillip.allen@enron.com
To: james.steffes@enron.com
Subject:
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Phillip K Allen
X-To: James D Steffes <James D Steffes/NA/Enron@Enron>
X-cc:
X-bcc:
X-Folder: \Phillip_Allen_Jan2002_1\Allen, Phillip K.\'Sent Mail
X-Origin: Allen-P
X-FileName: pallen (Non-Privileged).pst

Jim,

Is there going to be a conference call or some type of weekly
meeting about all the regulatory issues facing California this
week?  Can you make sure the gas desk is included.

Phillip
```

*Figure 1: Email from Phillip Allen to James Steffes*

The individuals' first and last names are not explicitly stated anywhere in the dataset, but most of them can be inferred from their email addresses. For this short-term project, it was decided to extract the text body, the "to" address or addresses, the "from" address or addresses, and the date. Had it been a multilingual dataset, the encoding or charset would have been very much worth extracting as well for any sort of text-mining purposes.

## Technologies Used

In order for this data to be stored in an RDBMS, Oracle 12c was selected as the DBMS platform, but this does not mean that it is the only or necessarily the best technology to utilize. Since this dataset is not a transactional database, and therefore eventual consistency would perfectly suffice, it would potentially be more useful to store it in a semi-structured database such as MongoDB. Moreover, the scalability would potentially greatly improve[5] the speed of querying large chunks of the email dataset. That being said, the data logically resides in a relational model that is easily understandable to the end user, with minimal difficulty in the variety of possible queries. Since it is not *big* data, only approximately 6 GB, Oracle is a very familiar and efficient enough technology to utilize for the experiment and remain within the scope of this course. The database was created on an Oracle 12c Docker image that was publicly available[6].

To populate the database, the files first had to be parsed. This required an object-oriented programming language that could read in the data, store it in cache as objects, and connect to the database and insert them.

Python was selected as the programming language, with the cx_Oracle module to connect to the database, for a few reasons. Its object declaration syntax provides flexibility for datatypes. Also, the iPython Notebook is a simple tool to use when performing trial-and-error experiment when working with larger datasets with variable anomalies. Instructions written and made publicly available by the Computer Science department at University of Texas at Austin[7] were used to set up cx_Oracle on the OS X local machine.

[5] Schudy, Module 6 Lecture Notes, BU MET CS 779 Spring 1
[6] https://hub.docker.com/r/sath89/oracle-12c/
[7] http://www.cs.utexas.edu/~mitra/csSpring2012/cs327/cx_mac.html

## Parsing the Files

As seen in Figure 1 (above), the text files were organized such that the metadata was at the top, with its label preceding it and separated by a colon. The text body occurs after the field "X-Filename". These two consistencies were found in all of the handful of randomly selected files, and therefor were utilized in parsing the data. It was stored in Python objects named Email, Person and Folder, and their attributes contained the metadata deemed worthy of utilization in this experiment. The usernames, or handles, as labelled in this project, were stored as the folder names representing individual owners of the different emails and folders. Because of this, the data was first stored as string objects located in an embedded Python Dictionary, where the first level keys represented Persons, the second level keys represented Folders, and the third level contained the strings read from the text files. A recursive method was utilized to capture the filenames as well as the files themselves. This Dictionary was further processed into objects with attributes that could be stored into the database using simple Regular Expressions.

## Database Design

Below is the schema design for the database.

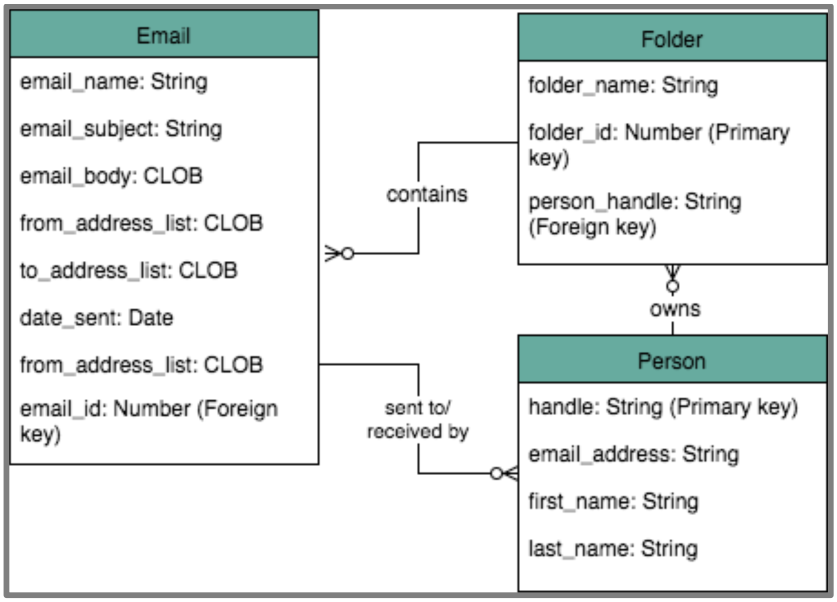


*Figure 2: Relational schema to contain the organized data*

As you can see, the relational model is simple and logically understandable by the average layman. The relationship between Person and Email, however, is not explicitly stated in the Database Design. The Person can be connected using the 'LIKE' keyword in Oracle to compare the strings. In future, the "to" and "from" address lists can be parsed out and connected to Persons. Alternatively, they could be further processed and contained by an array of the String objects. Currently, it is a CLOB object capable of holding address lists for emails with a massive audience.

## Populating the Database

Rather than inserting the data by uploading it directly the database in structured format as, for example, CSV or Excel files, the data was inserted by integrating Python with Oracle using the cx_Oracle module. This cut out the process of formatting files to upload, as well as the risk of crashing the SQL client on the local machine by trying to upload a file containing 500,000 emails at once. The upload took approximately 2 hours and ran without issue after including a try-catch block that accounted for the anomalies in the Date field.

## Querying the Data

Once the data was inserted, the data was queried for a sanity check, as well as to scratch the surface of its analytics capabilities. Some of the statistics of the database can be seen below from some of the queries:

```
Number of emails: 490706
Number of non-null email bodies: 490245
Number of non-null dates: 490245
```

These statistics indicate an extremely high insert rate, showing that the dates and the text bodies were almost always identified and inserted. The identical number for the number of non-null ema il bodies and non-null email dates indicates anomalous files that need to be further researched.

Some other tests were performed to test the ability to efficiently query the data. The below figure shows a wordcloud generated from a randomly selected handful of 5,000 of the emails.

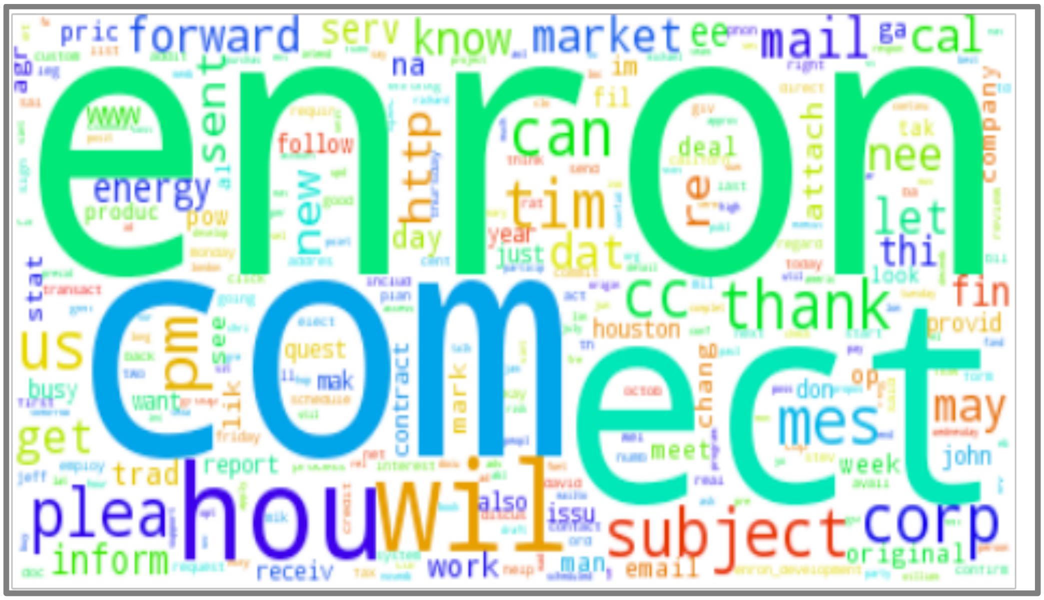


*Figure 3: Worcloud from 5,000 randomly selected emails*

Figure 4 below shows the results of the query selecting the top 10 Persons with the highest email activity.

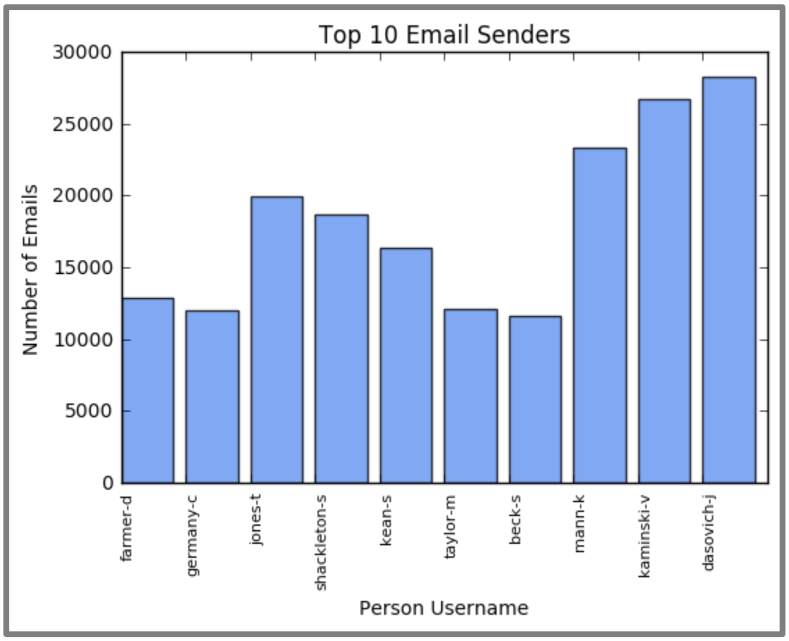


*Figure 4*

## Challenges and Future Directions

While the outcome of this project was an extremely useful tool that can be utilized not only for research specifically on the Enron emails but also as a text mining sandbox for similar projects, there are still a number of improvements that can be made to further improve the project.

The first noticeable improvement is the potential to extract further metadata. Firstly, the first and last names of the Persons can be extracted from most of the email sets. For the other emails not adhering to the same format, there is some room for experimentation with Computational Linguistics and Machine Learning to extract the names from the bodies of text themselves. Oftentimes, the email addresses only contain the full last name, but the first name can be found in many of the introductory salutations in email text. The emails attributed to individuals whose names were able to be extracted from the email addresses should be used as training data, and this training set could then provide a best guess to capture the names of the others. If the last name is not found in the email address, then the last name could be extracted by identifying di-grams with the first entry equal to the first name of the individual, and combine this with named-entity recognition to find the most likely candidate for the user's last name. This is an entirely separate project in itself which would require some more research and algorithmic design.

The email lists also need to be further organized. Currently, to relate an email directly to all of the individuals in the "to" and/or "from" address lists, they have to be identified via string matching. While this is not a terrible necessity, it would be further improved for gaining summary statistics and performing network analysis if these lists were organized into arrays of values, or if the database schema were further normalized to store these instances.

Some of the date and email body fields were anomalous, so there could be further research on these outliers to attempt to insert more of them into their respective cells.

The other metadata could be added to the database as well if a need were established.

## Conclusion

The methods outlined above provide a useful and reliable strategy for the ETL of similarly organized text and/or email datasets that can be performed by a data-oriented IT professional with an intermediate level of knowledge of available technologies and database logic and functionalities. While there is still some work ahead for this project, the limited timeframe in which the majority of the project was completed proves its effectiveness and viability for future projects.

## Bibliography

1. Berkooz, G. (2017, February 16). How Chief Officers Can Get Their Companies to Collect Clean Data. *Harvard Business Review*. Retrieved from https://hbr.org/2017/02/how-chief-data-officers-can-get-their-companies-to-collect-clean-data
2. Carnegie Mellon University Computer Science Department (2015*). Enron Email Dataset* [Data file]. Retrieved from https://www.cs.cmu.edu/~./enron/
3. Johnston, D. (2002, January 16). ENRON'S COLLAPSE: THE INVESTIGATION; Justice Dept.'s Inquiry Into Enron Is Beginning to Take Shape, Without Big Names. *New York Times*. Retrieved from http://www.nytimes.com/2002/01/16/business/enron-s-collapse-investigation-justice-dept-s-inquiry-into-enron-beginning-take.html
4. Schudy, R. *Module 6* [PDF document]. Retrieved from Lecture Notes Online Web site: https://onlinecampus.bu.edu/bbcswebdav/pid-4614173-dt-content-rid-16038004_1/courses/17sprgmetcs779_o1/module6/allpages.htm
5. Bilenko, M. (2003). Oracle Standard Edition 12c Release 1 [Software]. Available from https://hub.docker.com/r/sath89/oracle-12c/
6. (Spring 2012). *Build and Install cx_Oracle on Mac Leopard Intel.* Retrieved from http://www.cs.utexas.edu/~mitra/csSpring2012/cs327/cx_mac.html