# Sentiment Analysis of Emails

## Introduction

Email logs have been considered as a useful resource for research in fields like link analysis, social network analysis and textual analysis. This dataset is very similar to the kind of the data collected for fraud detection and counter terrorism hence it is a perfect test bed for testing the effectiveness of techniques used for counter terrorism and fraud detection. I plan to use this database for sentiment analysis of emails. In this report I describe my efforts to clean the data and how I propose to move forward from here.

## Enron Dataset

The Enron email dataset was made public by the Federal Energy Regulatory Commission during its investigation. It had a lot of integrity problems. It was later collected and prepared by Melinda Gervasio at SRI for the CALO (A Cognitive Assistant that Learns and Organizes) project; most of the integrity problems in the dataset had been resolved. It contains all kind of emails personal and official. Some of the emails have been deleted as part of the redaction effort due to requests from affected employees. William Cohen from CMU has put up the dataset on the web for researchers (http://www-2.cs.cmu.edu/~enron/ ). This version of the dataset contains around 517,431 emails from 151 users distributed in 3500 folders. These messages don’t include attachments. The dataset contains the folder information for each of the 151 employees. Each message present in the folders contains the senders and the receiver email address, date and time, subject, body, text and some other email specific technical details. Figure 1 shows the Mysql structure of the data

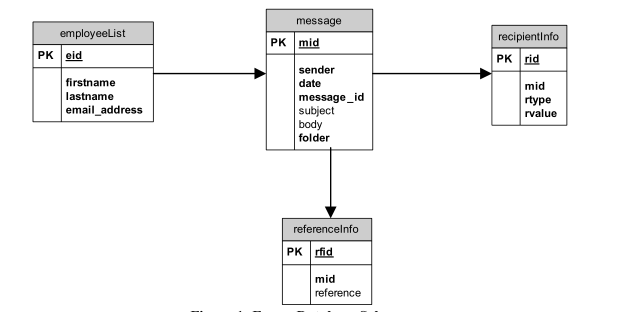


Figure 1

For initial exploration of the data I am only considering the email text and the ID. Later, I will be adding the email sent sender, receiver, attachments. A sample data is shown in figure 2.

|  |  |
| --- | --- |
| ID | Email text |
| 1 | System Notification: At 0115 PST, WACM terminated request for coordinated operation controllable devices for Path 30 USF mitigation. |
| 2 | Hey there Bill!  I thought I’d drop a quick line just to do a quick introduction of myself.  Your buddy, Brendan, has offered your tour guide ability to me when I visit  Portland in July. I do have a college friend out there (who I hope to locate) so you might not have to be stuck with me. ☺ |

Figure 2

I am planning to follow the following workflow for the project-

1. Data cleaning
2. Feature vector generation
3. Classification
4. Predictive modelling

## Data cleaning

1. In this process I am using the python to read the data.
2. Tokenize
3. Lowercase the entire data.
4. Word counts.
5. Remove common words.
6. Using nltk library I am removing the stop words and generating filtered data.

## Feature Vector Generation

### Dictionary generation

#### Bag of word Approach

I am generating a basic dictionary by find the unique words in all of the data and then removing the most common words such as “enron” and “subject”.

#### Word to vec

I plan to use the word to vec library to generate the features from the data.

#### Latent Dirchlet Allocation (LDA)

As an alternative I am also exploring to generate the list of topics form the email and use that features.[1]

#### Existing Sentiment analysis dictionary

Some of the online available sentiment analysis libraries are-

1. MPQA Opinion Corpus - The MPQA Opinion Corpus contains news articles from a wide variety of news sources manually annotated for opinions and other private states (i.e., beliefs, emotions, sentiments, speculations, etc.)[2].

## Classification

Based on the topics generated by the LDA I will like to classify the text and see the results.

## Predictive modeling

Based on the features generated by the word to vec or basic bag of word approach I plan to label the data. But I haven’t been able to figure out how to automatically label the text data.

Then I plan to try some common classification algorithms like SVM, but finally I want to do the classification in caffe,

References-

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. The Journal of Machine Learning Research, 3:993–1022, 2003.

[2] http://mpqa.cs.pitt.edu/