RNN Deep Neural Network Learning with Long Short Term Memory: Email Auto-Compose project

Ma Jia, Samuel Yang, Ying Liu, Yuanzhe Li, Yuhua He

San Jose State University, San Jose, CA, 95192, USA

**Inspired by Google’s Artificial intelligent tool called Smart Compose and the emerging chat-bot technologies, we want to come up with a deep learning solution to let robots help compose our emails.**

***Index Terms*—Natural Language Processing, RNN, Logistic Regression, Linear Regression, Deep Learning, Neural Network, Cross Validation, Machine Learning, ReLu activation function, Optimizer, Long Short Term Memory**

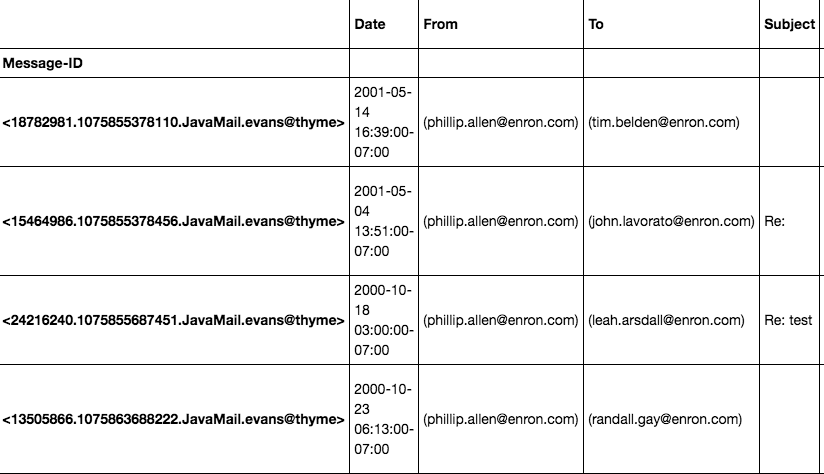
# Introduction

*In* this group assignment, the business objective we are focusing on, is the "Smart Email Compose" project. During the data summarization, we are using the Enron email dataset, which has the features like this: Date, From, To, Subject, Content, and so on. We decided to do to main things on this project. First, we gather sentiment information to analyze if there polarity of each mail relates to the stock price change of Enron Company in same period. Second, we will use RNN along with Long-Term-Short-Memory method to perform word generations, or text predictions.

# DATA Set, Scraping and Enrichment

The primary dataset in our project is the Enron email dataset from Kaggle. This dataset has email communications archive around 2002 before Enron scandal occurs. This dataset has 2 columns of Message and File. We parsed the data in the Message file to have multiple features to analyze. The secondary dataset we are using is the robot and human conversation dataset from a real-life chatbot called rDany, which is used for Telegram, Kik and Messenger’s custom service. The third dataset for enrichment is from New York Times url. We used beautiful soup to scape the Enron event from this website.

Before reducing noisy data, we decided to perform data enrichment to up-sampling our data. Since the dataset size is gigantic, we want to apply Latent Dichrilet Analysis(LDA) for categorizing them into ten topics. Afterwards, we will do Sentiment Analysis to the data enriched by the other two datasets (message chatting and Enron events) . The screenshot below is the enriched data.

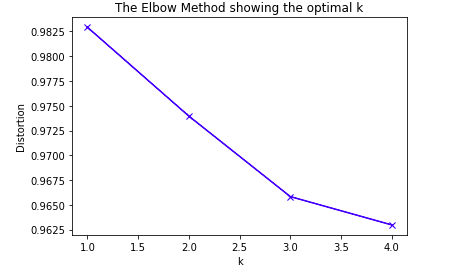
****

# IMPLEMENTATION DETAILS

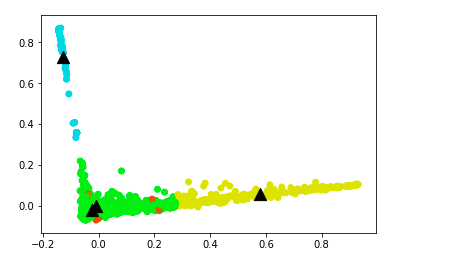
As for the data preprocessing, we used stemming and regular expressions to clear out stop words, titles, and other noisy data. The enriched dataset doesn’t have outliers and missing data. During the data transformation, we used LabelEncoder and bag of words method to convert the text data into numerical vectors. This helps run K-means and GMM. According to the elbow method, the ideal k value for K-means clustering is four.

**Unsupervised model:**

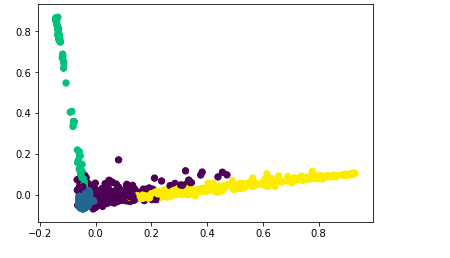
Elbow Method:



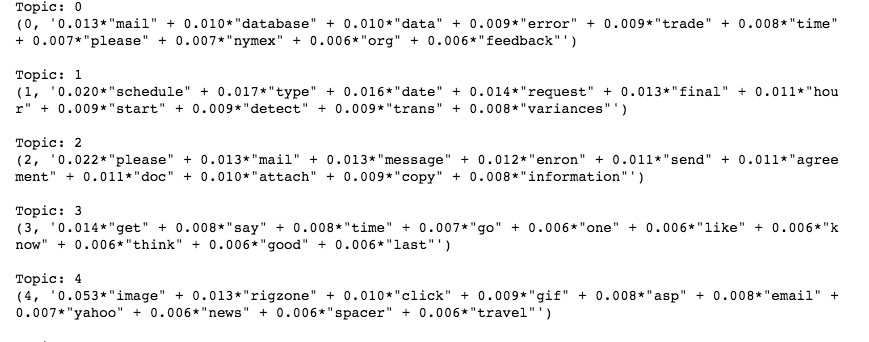
K-means



GMM:



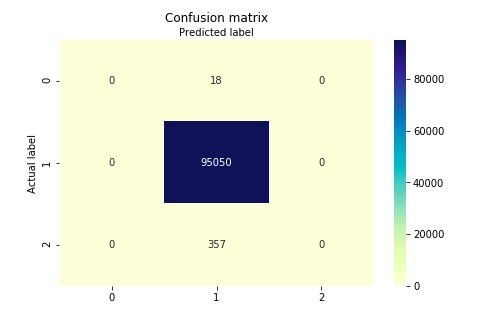
LDA:

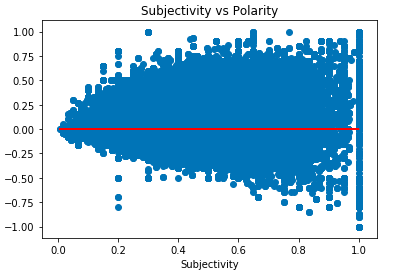


**Supervised Model:**

Linear Regression

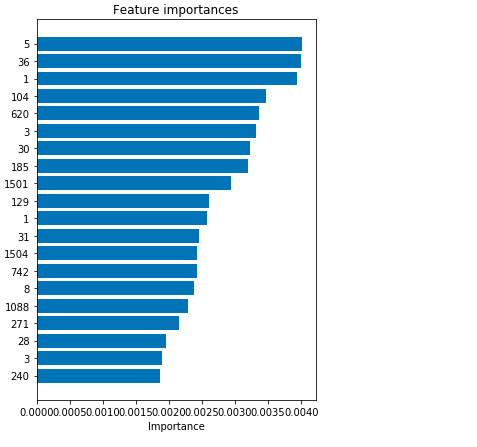
Logistic Regression and its confusion matrix:





**Embedded Model:**

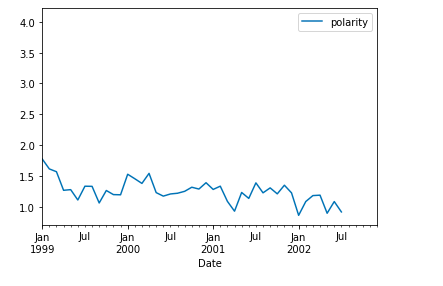
Random Forest: We uses RF to improve the accuracy and also prevent overfitting as well.



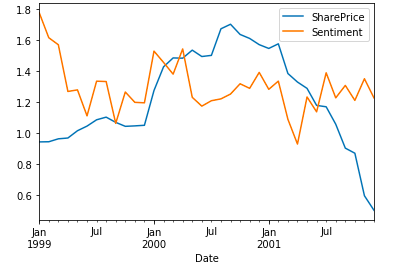
So now we are going to see the correlations between the amount of emails, sentiments, and the share prices.

Sentiment Analysis:

Sentiment Distribution:



Stock Price vs Sentiment



From the Sentiment distribution and amount of emails sending in particular period, and adding the Share Price into these data. In conclusion, from all the data analyses above, there might be indications that during the Enron period there might be some unusual activities and emails corresponding to the change of the share price. When share price changes, the amount of emails in that period also goes up a lot. As a result, they might secretly have some businesses behind that we don’t know.

For the auto complete model. We trained a model that is able to generate a 50 words email, based on given Enron email data set. Training such a big model is time consuming, even with GPU acceleration, we it still cost us more than 7 hours. Our result is showing below(Defined in Copy\_of\_email\_genarator.ipynb):

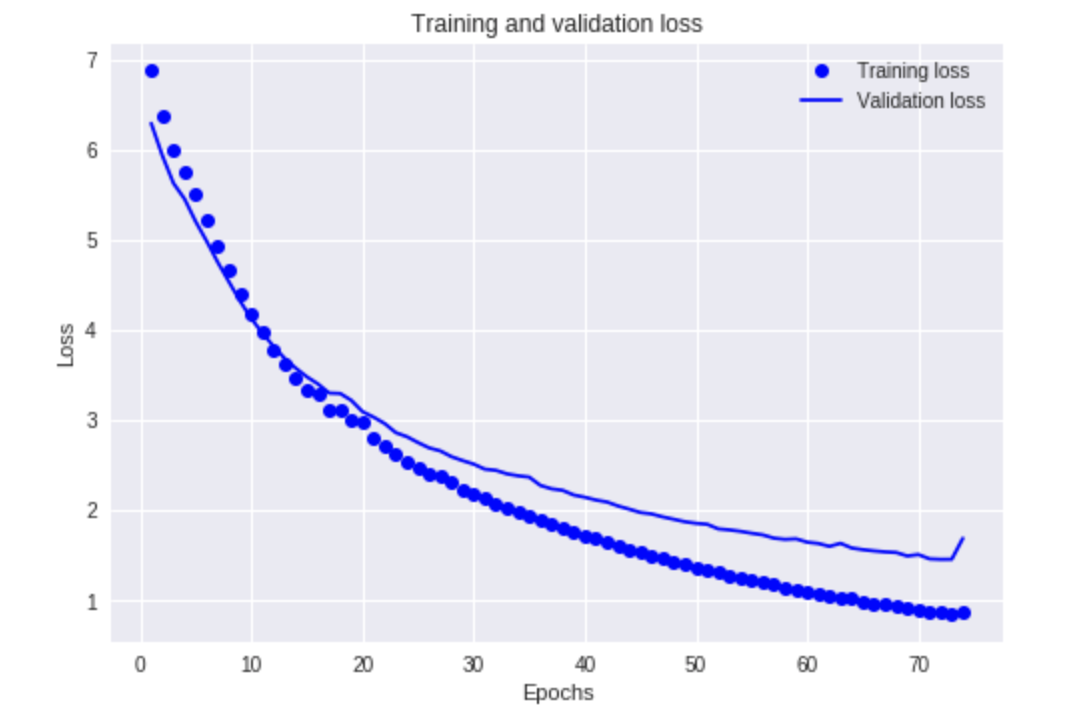
This is the training and validation loss chart during training.

For the email generator(test\_model.ipynb):

As we can see here, give a random selected text slice from original email data set (sequence pattern), it can generate a new text base on given content. Like the seed text is about give a feedback ASAP, so the generative text is about how to make a response.

RNN LSTM Model:

The input is the tokens of the original message(encoder RNN ), and the output is the conditional probability distribution of the sequence of response tokens given the input(decoder RNN). First, the sequence of original message tokens, including a special end-of-message tokens are read in, Then, given this hidden state, a softmax output is computed and interpreted as the probability distribution for the first response token. As response tokens are fed in, the softmax at each timestep is interpreted as the probability distribution for the first response token. Given the factorization above, these softmaxes can be used to compute conditional probability distribution.

**Reference:**

1.<https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45189.pdf>

2. <https://github.com/harvardnlp/seq2seq-attn>

3. <https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>

4. <https://ai.googleblog.com/2018/05/smart-compose-using-neural-networks-to.html>

5. Character-based Neural Machine Translation

<https://aclweb.org/anthology/P/P16/P16-2058.pdf>

6. Sequence to Sequence Learning with Neural Networks

<https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>

7. Smart Reply: Automated Response Suggestion for Email

<https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45189.pdf>

8. Recurrent Neural Networks for Language Understanding

<https://www.isca-speech.org/archive/archive_papers/interspeech_2013/i13_2524.pdf>