Analysis and visualization of Hillary Clinton's email dataset

by

Murali krishna teja Kilari , krishnateja@pec.edu,u1006392

Vinitha Yaski, vinitha.yaski@gmail.com, u1009987

Rohan Kohli, rohan.kohli@utah.edu, u0878757

Project Link:

https://github.com/krishnateja614/dataviscourse-pr-hillaryclintonemails

Report Links:

https://github.com/krishnateja614/dataviscourse15-hw-kilari-muralikrishnateja/blob/master/hw5/proposal_Kilari_Yaski_Kohli.pdf

https://github.com/yaskivinitha/dataviscourse15-hw-yaski-vinitha/blob/master/hw5/proposal_Kilari_Yaski_Kohli.pdf

https://github.com/rohan-kohli/dataviscourse15-hw-kohli-rohan/blob/360eebd3c0173e7e8c9da7a450be7f3402413866/hw5/proposal_Kilari_Yaski_Kohli.pdf

Background and Motivation

The dataset we chose for the class project is the “Hillary Clinton emails dataset” and we intend to do Topic Modelling on the data using Latent Dirichlet Allocation (LDA) to gain interesting insights and then visualize them interactively. We selected this because not only it will be quite interesting in the current political landscape of the country and in the scenario of upcoming elections but also we believe that with efficient preprocessing techniques and data mining algorithms we can try and extract some fascinating insights from this dataset.

Project Objectives

Our primary objective is to visualize a large dataset of emails in such a way that the most relevant topics discussed in those emails are programmatically brought to the attention of the user. As the dataset in question is the Hillary Clinton email dataset, we expect quite a few people to be interested in our visualization and the results that data mining brings to light.

The visualization techniques we will use(stacked area graphs and word cloud) haven’t been implemented as part of our assignments and another objective is to design as many visualizations as possible in this course.

Data

The dataset we are going to use is publicly available and is hosted at www.kaggle.com/c/hillary-clinton-emails . Please note that a raw version of this dataset is released by the government initially which consists of PDF files. However, the dataset on Kaggle is a cleaned version of the government version and the entire data is distributed over different files. The file names and their description is given below:

Emails.csv: It is the largest of all the files. It contains the actual information about the emails like Subject, From, To, Cc, Body text, Sender Id etc.

Persons.csv: Maps the Id to a person's name

Aliases.csv: Maps the Id to the Alias used in the From and To email fields and the person that alias refers to

EmailReceivers.csv: Maps the Id of the email to the Id of the person that received the email

There are in total 20220 rows in the Emails.csv prior to any data cleaning and preprocessing and it is safe to assume that the actual number of useful emails to our analysis will not differ by a very large number.

NOTE: Please note that even though Emails.csv contains other columns such as Case Number, Document Number and Date released, they are not of much importance to our analysis and hence will not be used in the project.

Data Processing

This step is going to be crucial because we should first clean the dataset to remove the irregularities and to remove emails that are missing the features we care about such as From, To, Subject, Body text. Also, after this step we should do Topic Modelling to extract the topics that exist in her emails. In this step, the number of topics is basically a hyper parameter that needs to be tuned manually until we get a good mixture of topics and a good distribution of words in each topic.

We intend to take the initial number of topics as 20 and then try incrementing that number until we find a decent topic model. In order to perform Topic Modelling we plan to use Latent Dirichlet Allocation (LDA) algorithm in Python. Since, LDA is a bag of Words model we should also clean the body text of each email so that we remove any "stop words" (words with very high frequency) from the body. This will be done by the NLTK library in Python.

The LDA algorithm will take as input the number of Topics and a corpus of the text and then tries to find the topics. Each topic is basically a collection of coherent or similar words with each word having a probability which is the probability of that word given the topic.

For example, the word distribution of one of the topics let's assume Topic –K will be Iran-0.34, Syria-0.21, Afghanistan- 0.17, Pakistan -0.14 etc. Then we can say that Topic-K is giving information about foreign countries and that too particularly about "muslim countries" and hence we can label Topic-K as "Foreign Countries"/ " Muslim Countries" since the words associated to them have higher probability of appearing within the topic. In the same way if a topic, say Topic- J has the following word distribution . Clinton-0.43, Bill-0.27, Chelsea-0.15, then it is safe to assume that this topic is about her family and hence label it accordingly.

There is also another way we are thinking of trying to group emails into separate categories and that is by using K-means clustering Algorithm, which as the name suggests clusters the emails into k groups. We plan on using the combination of K-means and LDA if it gives better grouping of emails and hence better topic models.

All the data cleanup and preprocessing will be done using Python.

Visualization Design

DESIGN I : To be implemented

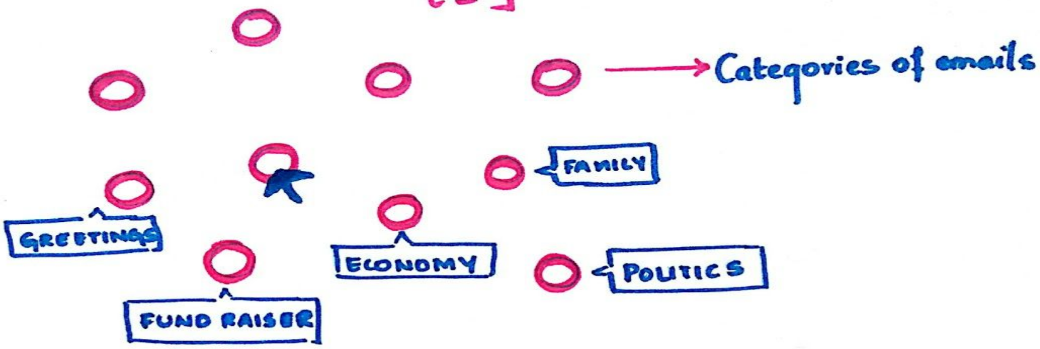
In this design, we have 3 visualizations and one email box that will be shown on the screen.

- The first visualization shown below, will display all the categories of emails in the form of circles. On click of any circle (or category), the most frequent words present in that category of emails are displayed using circles that are generated dynamically. These circles have their size and hue set according to their frequency in that category of mails
- The **second visualization** interacts with the first one, and displays a temporal stacked area chart of the words in the selected category. The X axis will have the period of time and this helps understanding which of the words are most frequent during any period of time.
- The **third visualization** is a word cloud that interacts with the first visualization. This displays the sender names (IDs) wherein the font size and weight are decided using the frequency with which Hillary Clinton interacts with them.

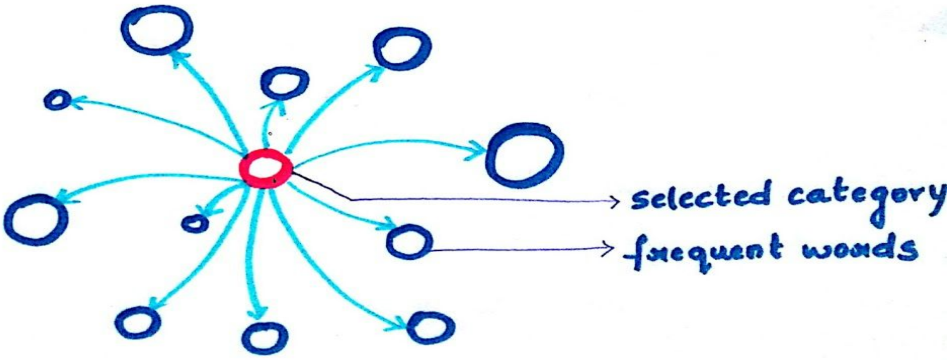
- To the end, there is a frame which will have the list of mails that are filtered using the selection made in any of visualizations.

VISUALIZATION -J
[I]

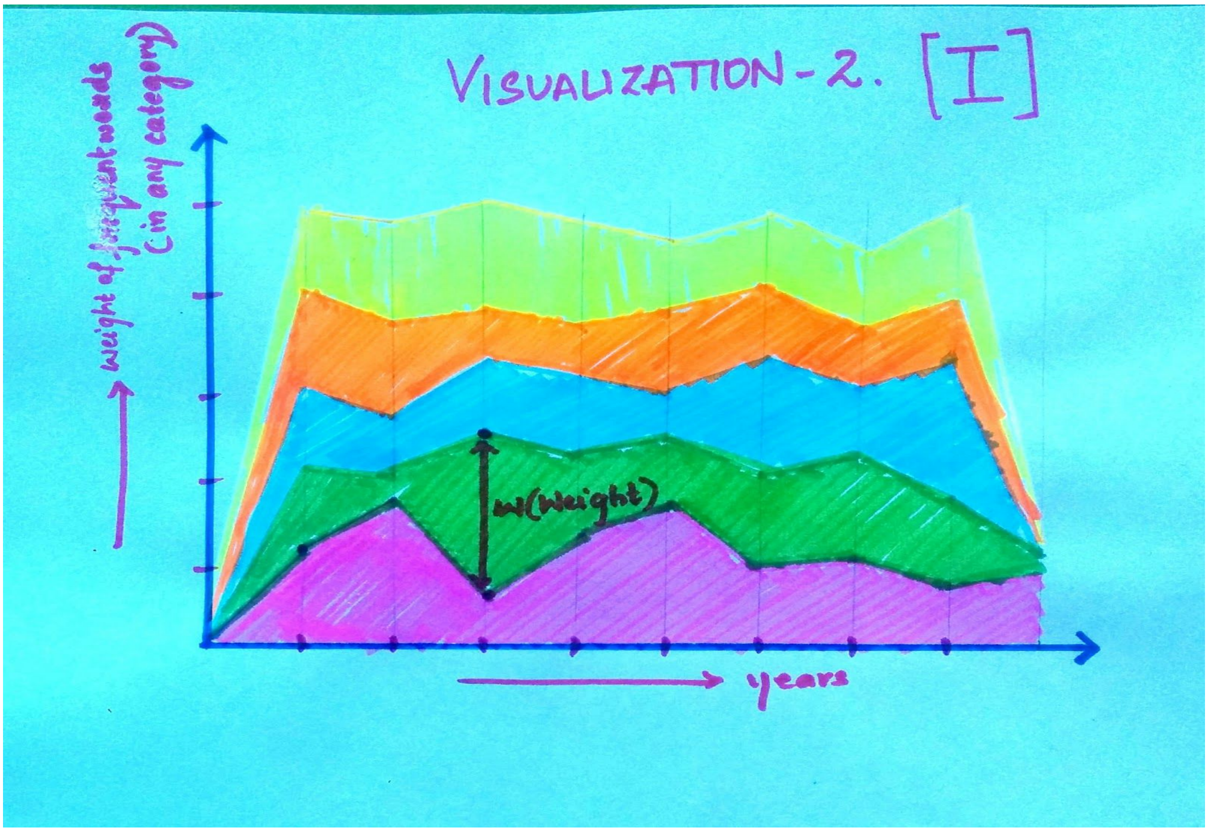
①



②



○↕ Size of circle and the saturation indicate the value of frequency



SENDER20
SENDER22
SENDER18
SENDER14
SENDER15
SENDER11
SENDER3
SENDER4
SENDER7
SENDER8
SENDER5
SENDER6
SENDER12
SENDER10
SENDER1
SENDER21
SENDER2
SENDER9
SENDER16

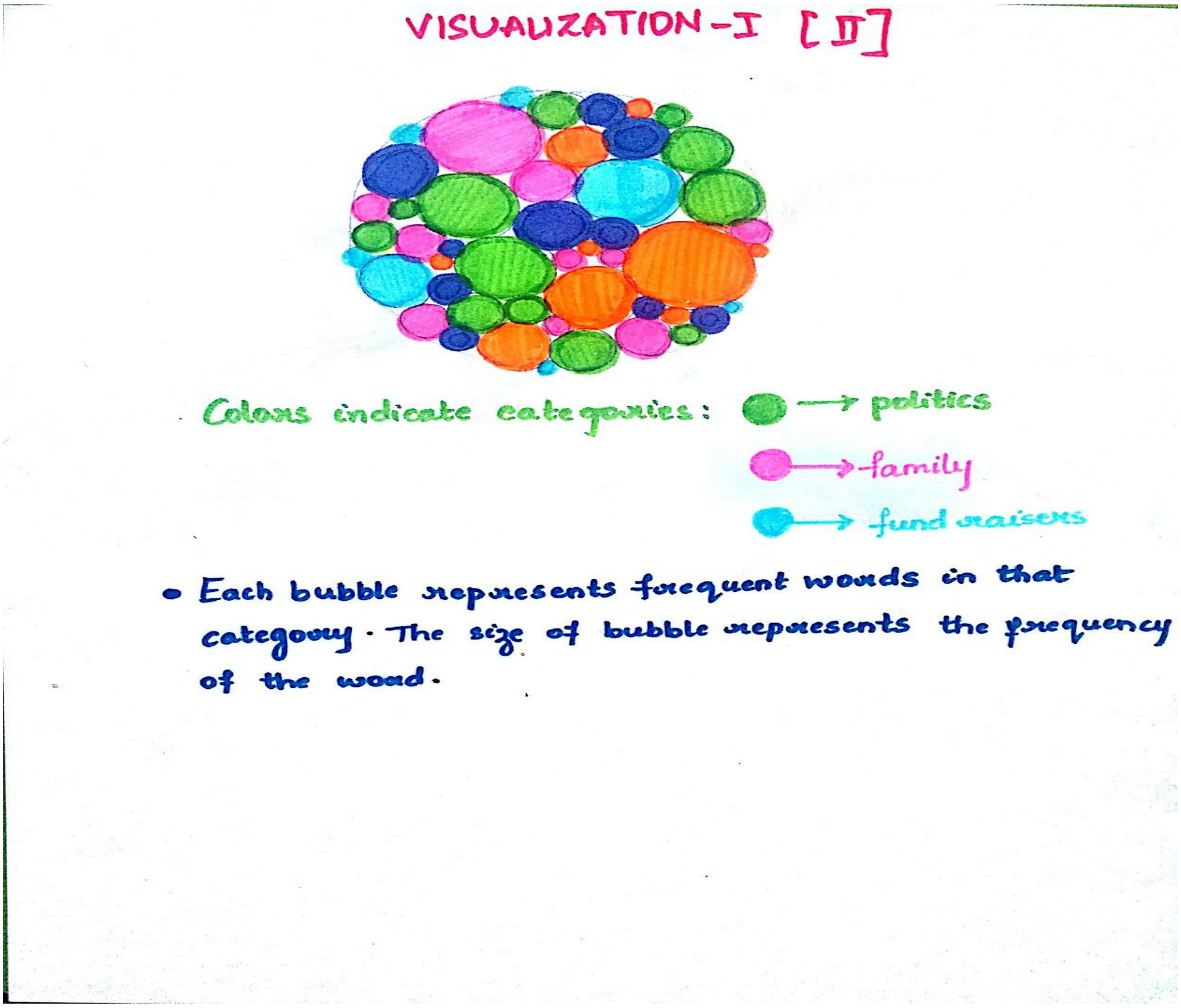
WORD CLOUD

VISUALIZATION-III [I]

ALTERNATE DESIGNS

DESIGN 2:

- The first visualization is a forced graph of 2D bubbles where each bubble represents the frequent words occurring in the mails and the color of the bubble determines the category to which they belong.
- The second visualization is stream chart of temporal data. This interacts with the first visualization and displays the most frequent words in a selected category. This is displayed in the form of streams whose height will be based on their frequency in any time range (marked on X-axis)
- The third visualization is a 'People Spiral' that interacts with the first visualization to display the people with whom Hillary Clinton interacts most frequently in the selected category. They are sorted according to their frequency.

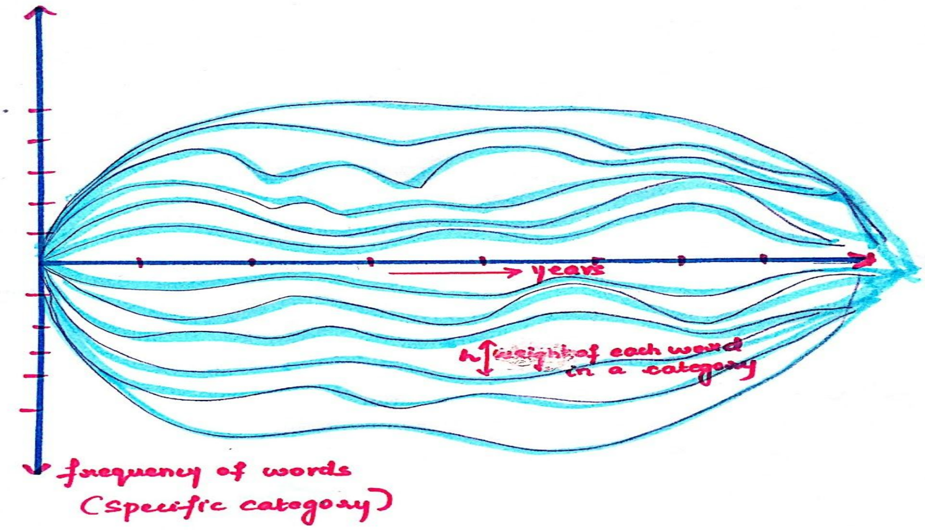


VISUALIZATION-I [II]

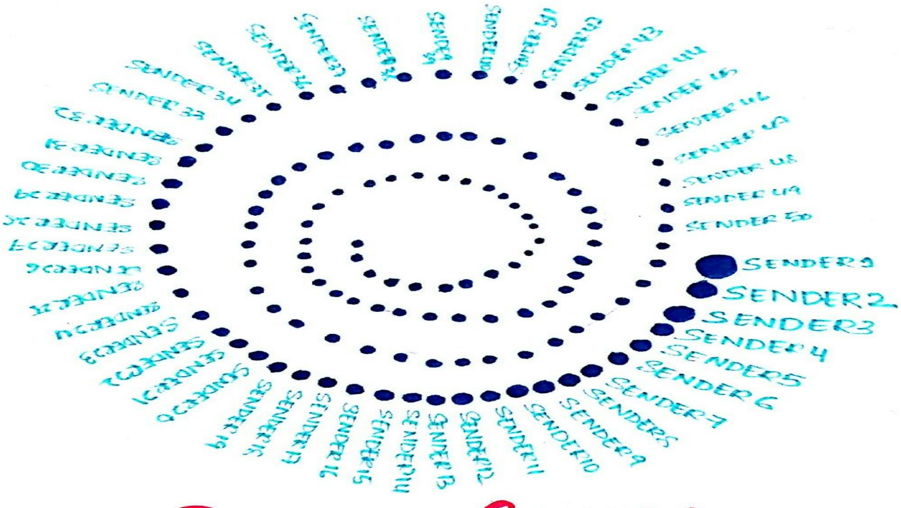
Colors indicate categories: ● → politics
● → family
● → fund raisers

- Each bubble represents frequent words in that category. The size of bubble represents the frequency of the word.

VISUALIZATION - 2 [II]



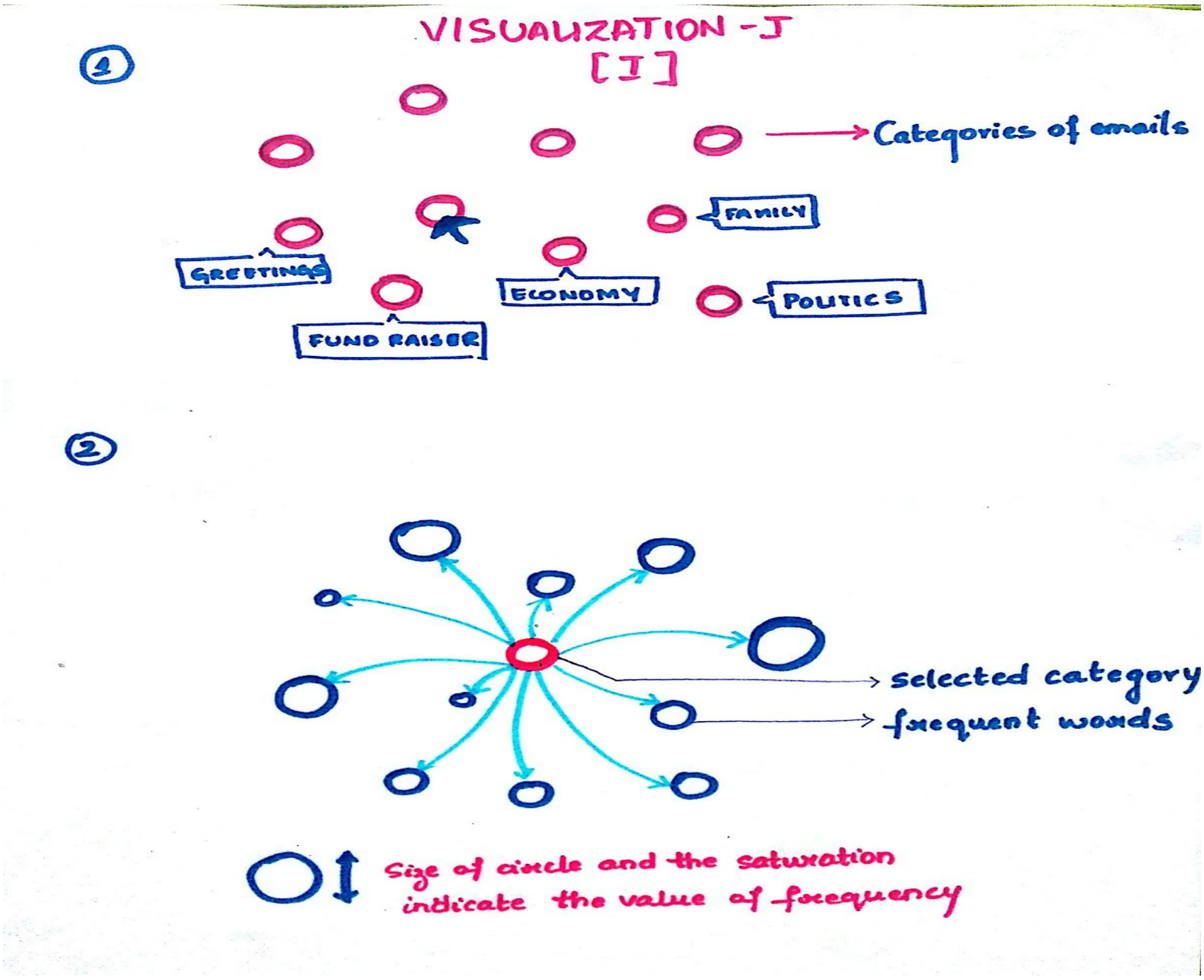
VISUALIZATION - III [II]



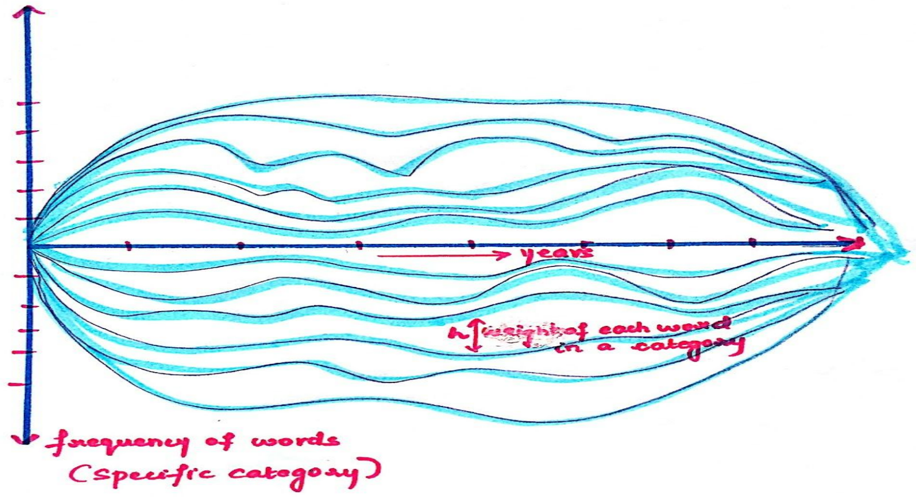
PEOPLE SPIRAL

Design 3:

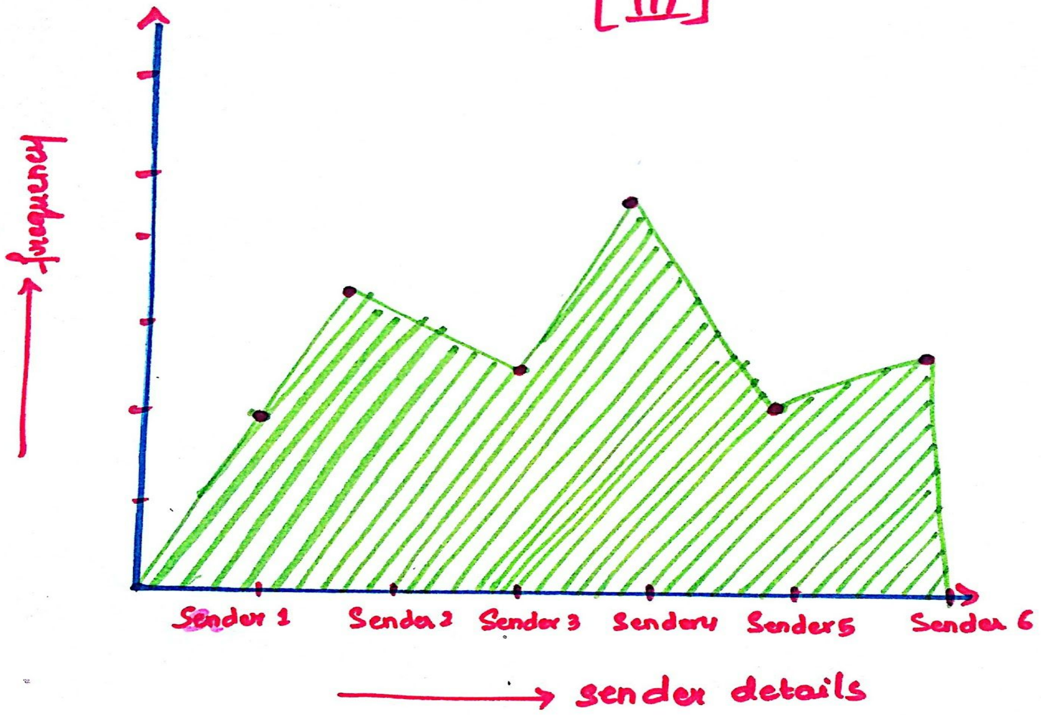
- The first visualization shown below, will display all the categories of emails in the form of circles. On click of any circle (or category), the most frequent words present in that category of emails are displayed using circles that are generated dynamically. These circles have their size and hue set according to their frequency in that category of mails.
- The second visualization is stream chart of temporal data. This interacts with the first visualization and displays the most frequent words in a selected category. This is displayed in the form of streams whose height will be based on their frequency in any time range (marked on X-axis).
- The third visualization is an area chart that responds to the selection made in the first visualization. It has the most list of people with whom Hillary Clinton interacts most frequently in any selected category. The Y axis shows the frequency of interaction.



VISUALIZATION - 2 [II]



VISUALIZATION - III [III]



Justification for the choice of designs:

The first visualization wherein we want to display the different categories of emails can be either represented as a free 2d bubbles that generate dynamic circles on click or using the force directed 2D bubbles. We have chosen to use the free 2D bubbles for the below mentioned reasons:

- As the count of categories of the emails increases, the force directed 2D bubble chart will have multiple colors and this will be an issue as the user will not be able to easily recognize the category to which a word belongs.
- The display of all the frequent words in all categories at the same time will be overwhelming to the user.

The second visualization for display of frequent words over a period of time can displayed using a stacked bar chart (or) stream chart. We have chosen the stacked area chart, for the below mentioned reasons:

- With the stream chart it would be difficult to identify discrete data at any point of time
- Also, taking the time constraint into consideration, it might be difficult to implement the stream chart as we need to use cubic bezier curves for the same.

The third visualization for the display of most frequent people who communicate with Hillary Clinton can be done using a Word cloud (or) people spiral (or) area chart. We have selected the 'word cloud' for the below mentioned reasons:

- The people spiral will be difficult to implement and it would be hard to visualize the details if there are too many people who interact in any category.
- Area chart will be easy for implementation, but it is not appealing to the users. Also, when the number of people who interact increases, the X axis will have too many details to display.
- Word cloud though does not help in identification of discrete details, it has a lot of visual appeal.We have chosen a combination of the best designs for each of the visualizations to get the efficient visual encoding for the data.

Must-Have Features

The project broadly consists of two sections: first, the pre-processing or the data mining performed to algorithmically create topics of interest in the email dataset, second, the four graphical elements used to visualize the dataset and topics of interest. The major features of our project necessary for its success are:

1. **Creating the topics of interest through data-mining**: the cluster of words in each topic must be closely related and provide a more meaningful result than a simple keyword search on the email dataset

2. **Implementing the four visualization techniques**: node graph, stacked area graph and the word cloud should map completely within their respective bounds. The text area listing the actual emails should limit the number of characters for each email to some constant and provide scrolling capability

3. **Interactivity and updates**: The four visualizations should be interactive whenever necessary and they should be interconnected i.e. changes in one visualization should update the others

Optional Features

There are some optional features which can greatly enhance the utility of our application, but given the time frame of the project, they are not necessary to implement:

1. **Alternate data mining algorithms:** Our current design uses Latent Dirichlet Allocation (LDA) for data mining. Although we can get different results using different parameters for LDA, it would be a nice feature to have multiple algorithms.

2. **Zoom in on stacked area graph:** A possible enhancement to our visualization is zooming in on the stacked area graph to reveal the content (emails), But we feel that implementing the graph as we designed it is complex enough so this may not be possible in 6 weeks.

1. **User can search for keywords:** The text section of our visualization (listing all the emails) isn't interactive in our current design. One way to increase its utility would be to add a textbox where users can search for emails containing particular words.

Project Schedule

The workload is divided as follows:

Murali Krishna Teja: Data cleanup and sorting emails according date and time, implement the data mining algorithm to generate topics of interest and provide them in JSON format to the web application

Vinitha Yaski: Create the html layout and CSS stylesheets, javascript prototypes for handling input data, implement static node graph and stacked area graph visualizations

Rohan Kohli: Implement the word cloud and text visualizations, make all four visualizations interactive, link and update all four visualizations with each other

WEEKLY DEADLINES:

WEEK 1: Clean and sort the dataset, remove any outliers. Create a shared repository, discuss and design the skeletal class structure for the application

WEEK 2: Implement Latent Dirichlet Allocation (LDA), the data mining algorithm we'll use, create the node graph and the word cloud visualizations

WEEK 3 (Project Milestone): Try different parameters for LDA to generate the most accurate result and return the result in JSON format to the web application. Populate the two visualizations using the results of data mining, link the two visualizations.

WEEK 4: Implement the remaining two visualizations

WEEK 5: Link all 4 visualizations and make them interactive. If possible, implement some of the optional features

WEEK 6 (Final Project Submission): Resolve any remaining bugs,create a write-up for the project website explaining how our application works, create the project screencast