PROCESS BOOK

TITLE: Analyzing and Visualizing Hillary Clinton emails dataset

Submitted by:

Kilari Murali Krishna teja

Vinitha Yaski

Rohan Kohli

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. We have a visualization of topics and the most relevant words in that particular topic. We also expect to have a stacked area chart that describes the temporal frequency of a word within a topic over the years. For this purpose, performing topic modeling than just counting the most frequent words that appear in the emails is clearly the better option as it gives us interesting clusters of words unlike the frequency count approach. 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.

Related work:

The following websites and papers discuss exciting ways to display textual and email data .We are thankful to the professor for pointing us towards them. The first paper actually inspired us to try and display the temporal frequency relationship using the stacked area chart.

1. http://mariandoerk.de/visualbackchannel/infovis2010.pdf
2. https://flowingdata.com/2008/03/19/21-ways-to-visualize-and-explore-your-email-inbox/
3. http://www.cs171.org/2015/assets/slides/12-TextVis.pdf

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

Id - unique identifier for internal reference

DocNumber - FOIA document number

MetadataSubject - Email SUBJECT field (from the FOIA metadata)

MetadataTo - Email TO field (from the FOIA metadata)

MetadataFrom - Email FROM field (from the FOIA metadata)

SenderPersonId - PersonId of the email sender (linking to Persons table)

MetadataDateSent - Date the email was sent (from the FOIA metadata)

MetadataDateReleased - Date the email was released (from the FOIA metadata)

MetadataPdfLink - Link to the original PDF document (from the FOIA metadata)

MetadataCaseNumber - Case number (from the FOIA metadata)

MetadataDocumentClass - Document class (from the FOIA metadata)

ExtractedSubject - Email SUBJECT field (extracted from the PDF)

ExtractedTo - Email TO field (extracted from the PDF)

ExtractedFrom - Email FROM field (extracted from the PDF)

ExtractedCc - Email CC field (extracted from the PDF)

ExtractedDateSent - Date the email was sent (extracted from the PDF)

ExtractedCaseNumber - Case number (extracted from the PDF)

ExtractedDocNumber - Doc number (extracted from the PDF)

ExtractedDateReleased - Date the email was released (extracted from the PDF)

ExtractedReleaseInPartOrFull - Whether the email was partially censored (extracted from the PDF)

ExtractedBodyText - Attempt to only pull out the text in the body that the email sender wrote (extracted from the PDF)

RawText - Raw email text (extracted from the PDF)

Persons.csv

Id - unique identifier for internal reference

Name - person's name

Aliases.csv

Id - unique identifier for internal reference

Alias - text in the From/To email fields that refers to the person

PersonId - person that the alias refers to

EmailReceivers.csv

Id - unique identifier for internal reference

EmailId - Id of the email

PersonId - Id of the person that received the email

database.sqlite

This SQLite database contains all of the above tables (Emails, Persons, Aliases, and EmailReceivers) with their corresponding fields. You can see the schema and ingest code under scripts/sqlImport.sql

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 be very different as the topic modelling algorithm requires atleast one non-sparse word in each email to take that email into consideration and hence emails with very short length will most likely be deleted by our preprocessing.

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 Cleaning and preprocessing:

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. Currently, since for topic models, we only consider Body text part of the emails, we describe in detail below, the steps that are performed in **R** to read, clean and preprocess the data.

- **Data reading** : The package "readr" is used in R to read the emails.csv file. Since, the size of this file is moderately large and most of the content is "string" this package will be more efficient than R's default csv reader.
- **Data cleaning and preprocessing**: Since, this is the actual emails dataset there will be a lot of noise in the content of emails and hence the following steps are performed on the dataset to clean it and make it more compact and useful. The "tm" package of R is used for all these steps

1. Creating a corpus: The “bodytext” column is extracted from the csv and a corpus is created
2. Lowering: All characters in the corpus are converted to lower case
3. Conversion the corpus to plain text document
4. Removing punctuations from the corpus
5. Removing numbers from the corpus
6. Removing whitespace from the corpus
7. Removing stopwords from the corpus
8. The corpus is converted into document term matrix and terms with sparsity less than 0.95 are removed from the corpus. If a word is highly sparse, then it means that it occurs very infrequently.

Data mining - Topic Modelling:

Also, after this step we should do Topic Modelling (“topicmodels” package in R) 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 number of topics to be 20 (NOTE: for the purpose of milestone,the code in the github repository will have the number of topics as 10 because of the huge memory and time taken by the algorithm) 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 R. 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 as done in step 7 of data cleaning and preprocessing.

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.

1. Interpretation of Topic models: Topic models discover “topics” that occur in a collection of text. Each topic has a different probability mass function over the set of words (all vocabulary). We want to discover which words are more relevant for which topic
2. Measure-1: Importance of words in a topic is denoted by the probability of the word given that topic. Let’s call it “**P**”
3. Disadvantage with “**P**”: common words tend to appear in every topic and hence will have high “**P**” values for every topic, thus making it difficult to efficiently and accurately discover and differentiate topics

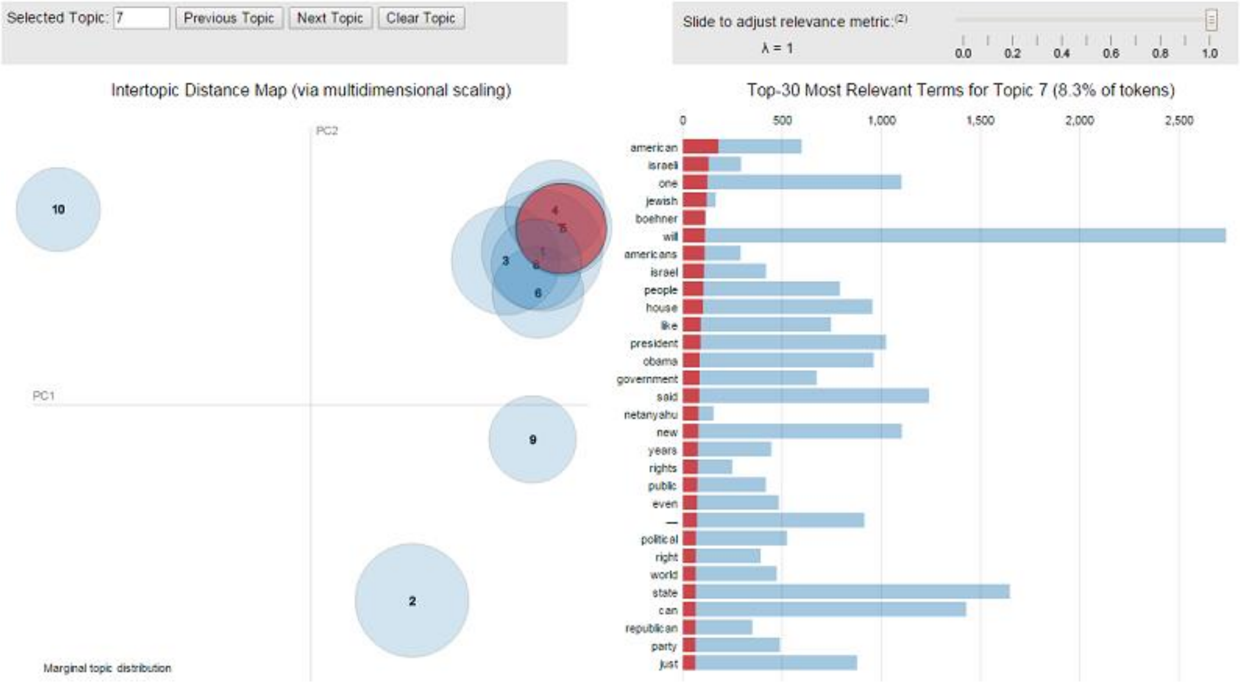
4. Measure-2: We can modify “**P**” by dividing “**P**” with overall probability of the word. Let’s name is “**L**”. “**L**” takes care of the above disadvantage.But,
5. Disadvantage with “**L**”: In the case of rare words (whose probability of occurrence is very low) “**L**” will have high values since we will be dividing “**P**” with very small values.
6. **RELEVANCE PARAMETER**: The relevance parameter **λ** is a tradeoff parameter between ”P” and “R”.

Relevance = **λ*P + (1- λ)*L**

Design:

The following three visualizations show the word distributions for topic “7” which is talking about American government, president Obama and israel for different values of **λ**

1.λ =1

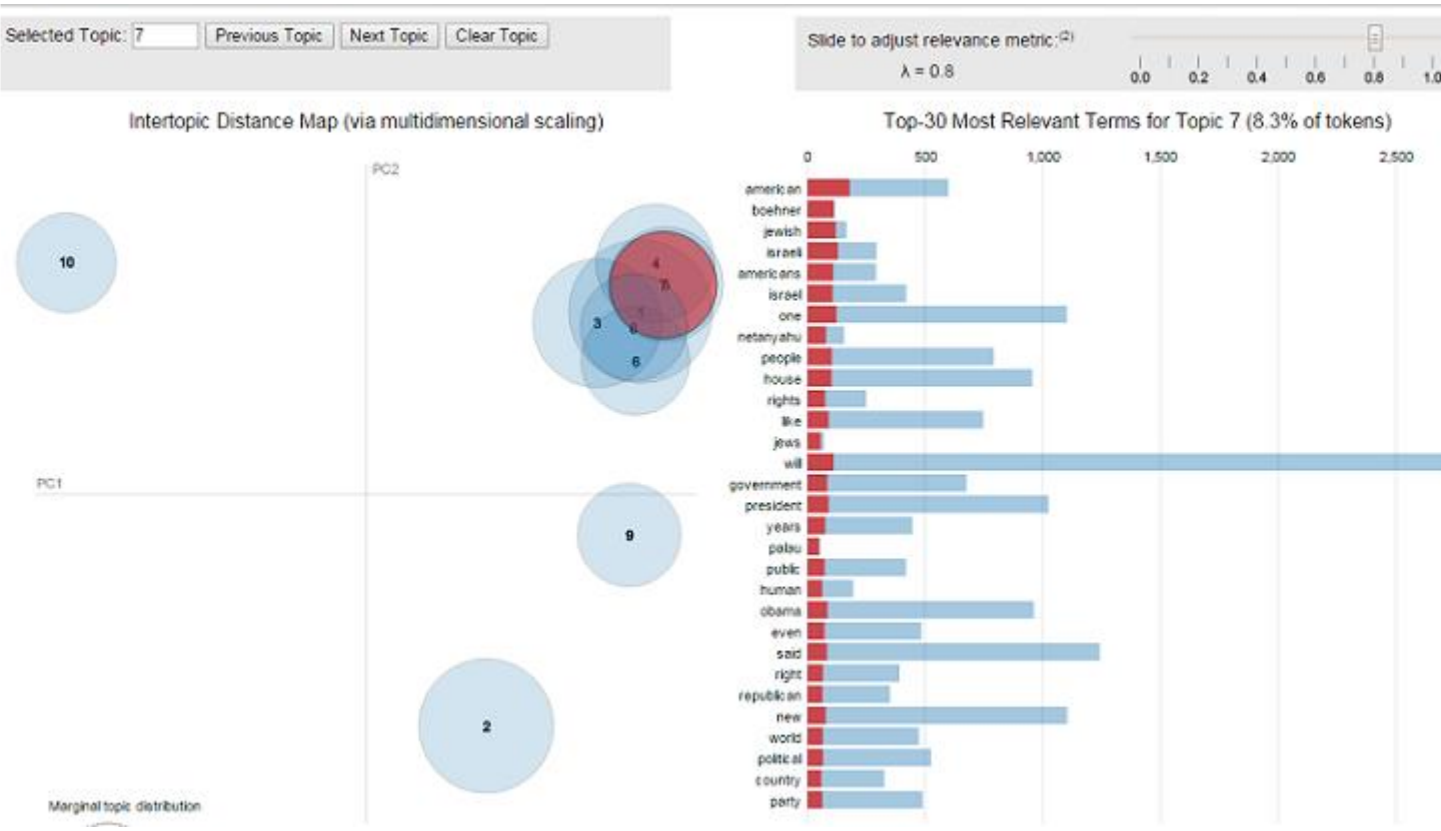


Selected Topic: 7

Slide to adjust relevance metric:(2)

λ = 1

2. λ = 0.8

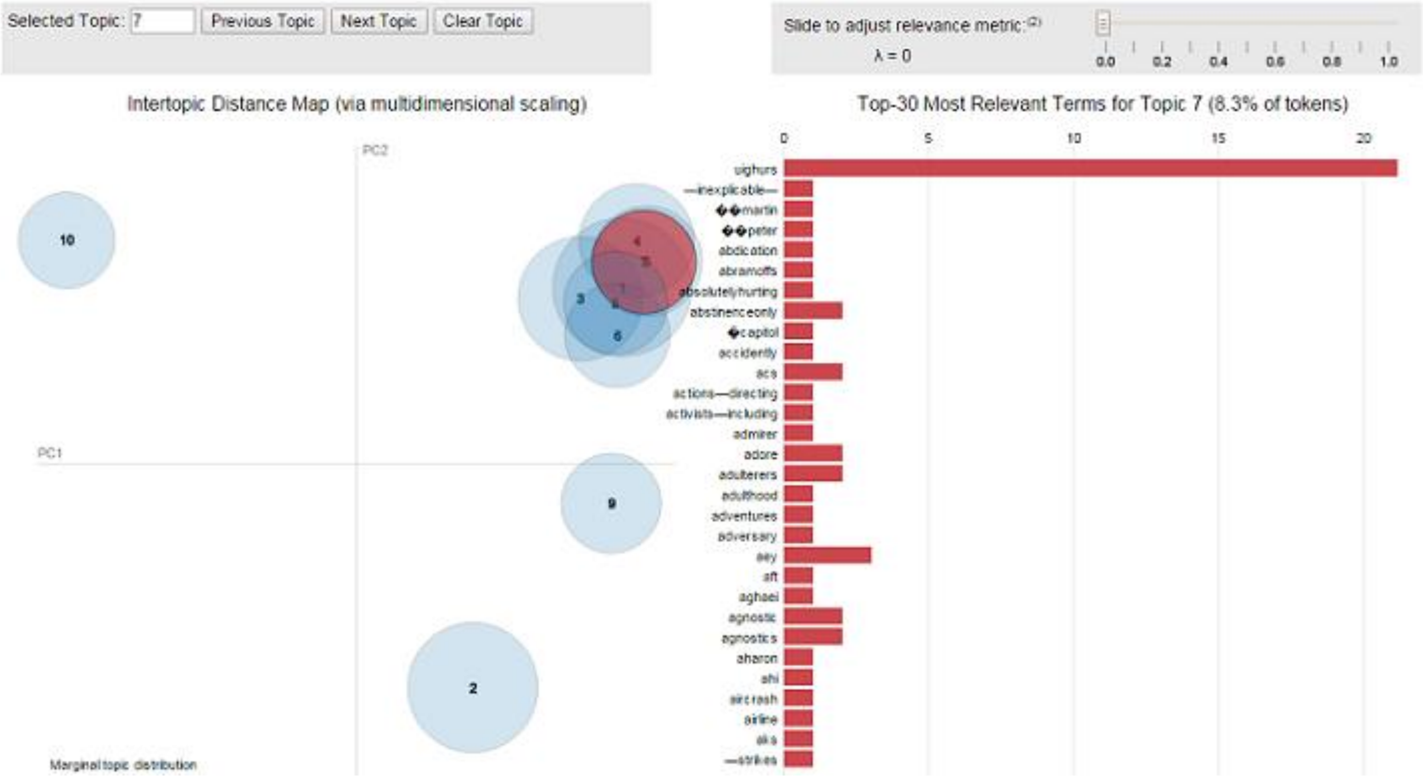


Selected Topic: 7

Slide to adjust relevance metric:(2)

λ = 0.8

3. λ=0



Selected Topic: 7

Slide to adjust relevance metric:(2)

λ = 0

IMPLEMENTATION:

In the project github repository, only the file named **“LDA Hillary.R”** is to be executed. All the other folders and files belong to another library named “LDAvis” which we are trying to edit**. Also make sure that both the “LDA Hillary.R” file and the data folder named “Hillary Clinton public emails dataset” are in the same directory and that this directory is the working directory of “R”.** Please follow the below steps to run the file in the order they are given

1. **Packages required:** Run the following commands in R console to install the required packages before running **“LDA Hillary.R”** file . You can copy/paste the following lines to install them.

   ```
   install.packages("readr")
   install.packages("topicmodels")
   install.packages("dplyr")
   install.packages("stringi")
   install.packages("tm")
   install.packages("LDAvis")
   install.packages("servr")
   ```

2. **Run the file by copy/pasting the entire file into the R console.**