Email Analytics

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Master of Technology

in

Computer Science & Engineering

with specialization in Big Data Analytics

by

Gajanan Rajendra Mirkhale

14MCB1054

School of Computing Science and Engineering,

VIT University, Chennai,

Vandalur-Kelambakkam Road,

Chennai - 600127, India.

May 2017

VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
www.vit.ac.in
Vellore ▪ Chennai
CHENNAI CAMPUS
Vandalur - Kelambakkam Road, Chennai - 600127

Declaration

I hereby declare that the dissertation ***Email Analytics*** submitted by me to the School of Computing Science and Engineering, VIT University Chennai, 600 127 in partial fulfillment of the requirements for the award of **Master of Technology** in **Computer Science & Engineering with specialization in Big Data Analytics** is a bona-fide record of the work carried out by me under the supervision of ***Prof. S.A.Sajidha***.
I further declare that the work reported in this dissertation, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Sign:

Name & Reg. No.: Gajanan R. Mirkhale, 14MCB1054

Date:

VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
www.vit.ac.in
Vellore • Chennai
CHENNAI CAMPUS
Vandalur - Kelambakkam Road, Chennai - 600127

School of Computing Science & Engineering

Certificate

This is to certify that the dissertation entitled ***Email Analytics*** submitted by ***Gajanan Rajendra Mirkhale*** **(Reg. No. 14MCB1054)** to VIT University Chennai, in partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering with specialization in Big Data Analytics** is a bona-fide work carried out under my supervision. The dissertation fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this dissertation have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Supervisor

Signature: .....

Name: Prof. S. A. Sajidha

Date:

Program Chair

Signature: .....

Name: Dr. Bharadwaja Kumar

Date:

Examiner

Signature: .....

Name: .....

Date:

(Seal of the School)

Abstract

In Email Analytics, our main focus on criminal and civil investigation from large email dataset. It is very difficult to deal with challenging task for investigator due to large size of email dataset. This paper offer an interactive email analytics various to current and manually intensive technique is used for search evidence from large email dataset. In investigation process, many emails are irrelevant to the investigation so it will force investigator to search carefully through email in order to find relevant emails manually. This process is very costly in terms of money and times. To help to investigation process. We combine Elasticsearch, Logstash and Kibana for data storing, data preprocessing, data visualization and data analytics and displaying results. In this process reduce the number of email which are irrelevant for investigation. It shows the relationship between them and also analyzing the email corpus based on topic relation using text mining.

Acknowledgements

I wish to express my sincere thanks to Dr.G.Viswanathan, Chancellor, Mr. Sankar Viswanathan, Vice President, Ms. Kadhambari S. Viswanathan, Assistant Vice President, Dr. Anand A. Samuel, Vice Chancellor and Dr. P. Gunasekaran, Pro-Vice Chancellor for providing me an excellent academic environment and facilities for pursuing M.Tech. program. I am grateful to Dr. Vaidehi Vijayakumar, Dean of School of Computing Science and Engineering, VIT University, Chennai and to Dr. V. Vijayakumar, Associate Dean. I wish to express my sincere gratitude to Dr. Bharadwaja Kumar, Program chair of M.Tech Big data analytics for providing me an opportunity to do my project work. I would like to express my gratitude to my internal guide Prof. S. A. Sajidha and my external guide Mr. Bharanetharan Sankaravadivelu who inspite of their buy schedule guided me in the correct path. I am thankful to Innova Solutions Pvt. Ltd.,Chennai for giving me an opportunity to work on my project and helped me gain knowledge. I thank my family and friends who motivated me during the course of the project work.

Contents

List of Figures

Chapter 1

Introduction

This thesis is written for the project which is under development known as **Email Analytics**.In Email Analytics large email dataset is received and generated. This Email data set is represent technique to discovery of evidence and information in investigation from a large email dataset. So in any large email dataset to prevent the investigator form conducting a manual search. There is a need to reduces effort and saves a lot of business time to automate such activities.

1.1 Email Communication

Email is most of your day to day communication today. The surprisingly fast acceptance of the communication medium. This form of communication has easy to use and costing virtually nothing per message. In the digital age, people use written communication far more than ever before. In fact, email communication is not only used instead of letter writing, it has also replaced telephone calls in many situations and in professional environments. In this book Visualization analysis and Design (Tamara Munzner 2015) Tamara Munzner given explanation about the visual analytics when the exact questions are not known. Email analytics ability to find the human pattern, trends and anomalies.it is very difficult to investigation when content of emails are change.

Email analytics also analyzing the email corpus based on topic relation using text mining. In text summarization a large collections of emails are transformed to a reduced and compact email dataset, which represents the digest of the original email collections. This can be done using topic modeling algorithm. A summarized email helps in understanding the gist of the large email corpus quickly and also save a lot of time by avoiding reading of each individual email in a large email corpus.

1.2 Objectives

Now a days investigation of email through keyword of both headers and contents of email using methods. Still we are unclear with best keywords for searching the emails in the result. This methodology provides the best keyword for searching the emails in the result. It also reduce the number of emails from large dataset. Data visualization provides relationships in the context of the data, finding human pattern, trends and anomalies easier. Topic modeling provides summarization a large collections of emails are transformed to a reduced and compact email dataset

1.3 Challenges of Email Analytics

First, the data sets are very large and are growing rapidly which provides a challenge to finding relevant information. Second, our interviews revealed a lack of a good set of investigative tools to deal with many of the issues created by large email data sets. These issues include:

- Reducing the size of keywords search results.
- Removing duplicate,irrelevant or unimportant emails from large email datasets.
- Discovering inconsistency in the email data.
- Inability to summarize search results or different subsets of emails data.
- Finding indirect connections between email accounts.

Currently, In the market right now there are no specific techniques or tools to automate this process. The main reason being that the problem which we have to solve is very specific to an organization. email analytics is not providing 100 percent accuracy but it will be enough to automate the process.

Chapter 2

Literature Survey

2.1 A framework for the forensic investigation of unstructured email relationship data

The most comprehensive work on text based data for investigative analysis has been done by the Jigsaw Project at Georgia Tech by Stasko et al. (Yi et al., 2007; Kang et al., 2011; Liu et al., 2013). Their work focuses on supporting the investigative process by creating tools that help the analyst find and map relationships found in data sets. These relationships can be between people, places and things in any combination. These tools help the analyst piece together a coherent story from information contained in a document set which is limited to several thousand documents. For emails this size limit is not adequate, InVEST adopts visualization techniques that find relationships in data sets numbering in the hundreds of thousands.

2.2 Forensic triage of email network narratives through visualisation

the main purpose of this paper is automates the visualisation of quantitative(network) and qualitative data (content) within email dataset. Nowdays,emails are key source for evidence during the digital investigation and investigator examine may be required to triage and analyse the large amount of data.currently,we utilizes tools and techniques are manual through such data.this process is a time consuming process.

2.3 InVEST: Intelligent visual email search and triage

The main purpose of this paper(Jay Koven(2016)) criminal and civil investigation.In large size of data,investigation usually contain many emails which are not related to an investigation.so investigator manually arrange through email data in order to find relevant emails.To aim of this investigation process to automate the introduction for reducing the number of emails in search result.using our technique elk stack ,investigation is faster and it is reduce time and cost both.

2.4 THREAD ARCS: An Email Thread Visualization

It shows the relationship between the sender and receiver in emails threads using a unique visualization are which displays collections between sender and receiver using a series of arcing arrows to showing the relationships. This work shows that importance of tracing the sender and receiver connections in search result to email threads. Using our paper technique to showing relationships between sender/receiver with email subjects and connects to give clearer picture of the data.

2.5 Text Preprocessing

Text processing is a important procedure when we manage with textual data. With a specific end goal to apply any examination on the content, we initially need to preprocess the information or data. For preprocessing of the information we utilize the Natural Language Processing (NLP) strategies.

- Tokenization
- Lower case conversion
- Stop word removal methods
- Stemming
- Removal of punctuation marks
- Lemmatization

2.5.1 Tokenization

It is a technique is utilized to split stream of content up into words, expressions, symbols or other important components called tokens. The list of tokens then created then moves toward becoming contribution for further preparing, for example, grouping, content mining or parsing.

2.5.2 Stop word removal methods

The stop words are the words which are to be expelled before doing any examination on the content. They are fundamentally the most widely recognized words in the English dictionary.

2.5.3 Stemming method

Stemming is the method of identify the root form of the word.

2.5.4 Lemmatization

Lemmatization in linguistics is the process of grouping together the different forms of a word so that they can be analyzed as a single item. In computational linguistics, it is the algorithmic process of determining the lemma for the given word.

2.6 Semantic Similarity

It is a metric defined over a set of documents or terms, where the main idea of distance between them is based on the likeliness of their meaning and semantic content. These are mathematical tools used to estimate the strength of the semantic relationship between units of language, concepts or instances, through a numerical description obtained according to the comparison of information supporting their meaning or describing their nature. Computationally, semantic similarity can be estimated by defining a topological similarity, by using ontologies to define the distance between terms/concepts.

Semantic similarity can discovers applications in many fields, for example, Biomedical Informatics, GeoInformatics, Normal Dialect Preparing, and so on.

The different measures by which we can compute Semantic Similarity are Jaccard index, cosine similarity, shingling, etc.

2.6.1 Jaccard Index

It considers the documents as a set of words. It finds an intersection in the sets and gives a value of similarity between the documents.

Consider the following example wherein we consider two sentences. We try to find the similarity between these two sentences.

A. The cat ate the fish and drank water.

B. Fish swims in the water.

Set A = the, cat, ate, fish, and, drank, water

Set B = fish, swims, in, the, water

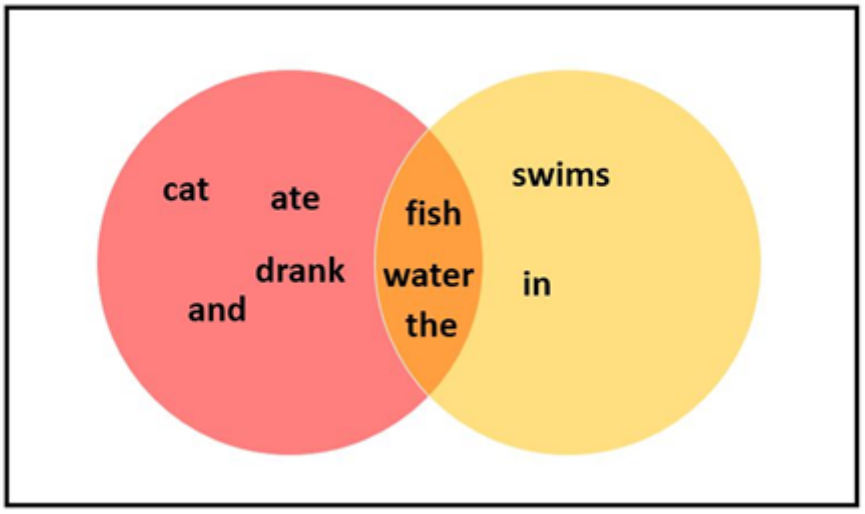

FIGURE 2.1: Set Diagram for finding Jaccard Similarity

formula : Jaccard Index is:

$$J(A, B) = (A \cup B)/(A \cap B) = 3/9 = 0.333 \tag{2.1}$$

2.6.2 Cosine Text Similarity

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. So more the value of similarity less will be the angle and hence the documents will be more similar to each other.

The basic working of Cosine similarity is as follows:

- Take the two documents which you want to compare.
- Remove the stop words from these two documents.
- From this figure the vectors of both the records or document.
- Take internal result of these two vectors. This will give the cosine of the edge between them.

formula : calculating the Cosine similarity is:

$$Cosine_sim(A, B) = cos(\theta) = A.B/||A||||B|| \tag{2.2}$$

2.6.3 Tf-idf Weighting

Term Frequency:
tft,d of term t in document d is defined as the number of times that t occurs in document d.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2.3}$$

Inverted Document Frequency:
Estimate uniqueness of a term in the whole documents collection. If a term occurs in all documents then its Idf is zero.

$$idf_i = log\frac{|D|}{\{|d : t_i \in d|\}} \tag{2.4}$$

Tf-idf:
The tf-idf weight of a term is the product of its tf (Term Frequency) weight and its idf (Inverted document frequency) weight

Example:
Consider a bank document containing 100 words where in the word miracle appears 3 times. The term frequency (i.e., tf) for miracle is then (3 / 100) = 0.03.

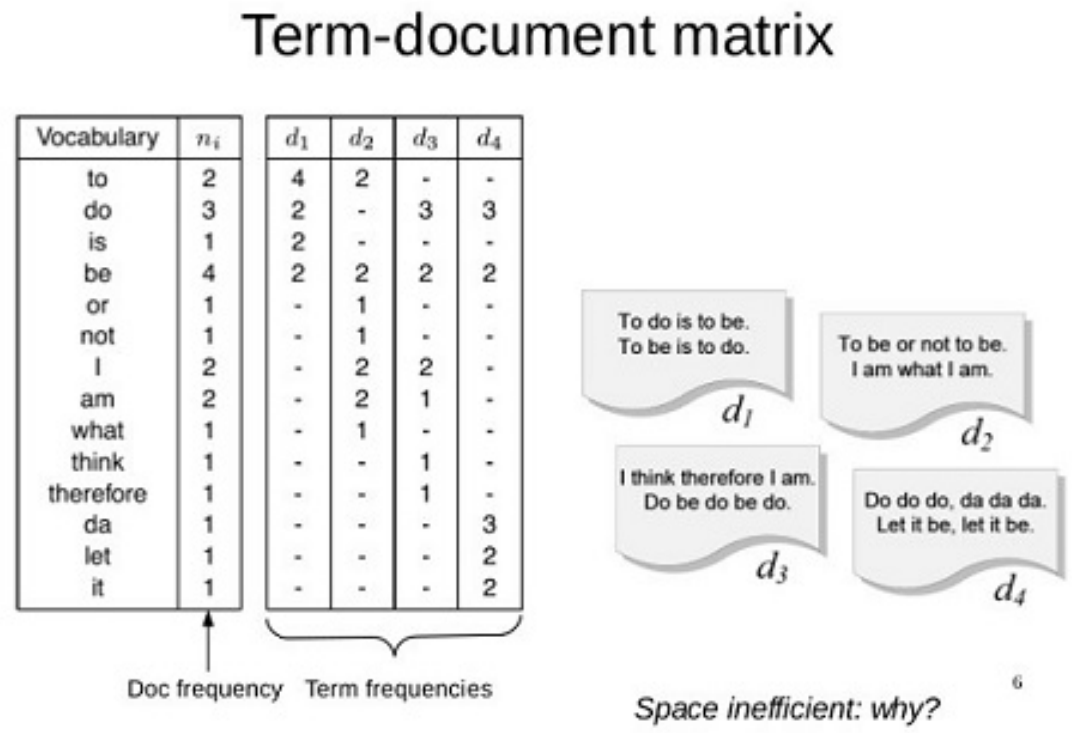

FIGURE 2.2: Diagram for Term document matrix

Now, assume we have 10 million bank documents and the word miracle appears in one thousands of these. Then, the inverse document frequency (i.e., idf) is calculated as log (10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.

2.6.4 Inverted Index

Inverted index is search engine indexing to collect, parses and stores data to facilitate fast and accurate retrieval of information. It is uses for full text searches. It is very easy to use, versatile and agile structure which provides efficient and fast test search capabilities.

An Inverted index consist of:

- It is appear list of all unique words in any document.
- The words appear in list of the documents.
- Term frequency shows that how many times a word has occurred in the list.

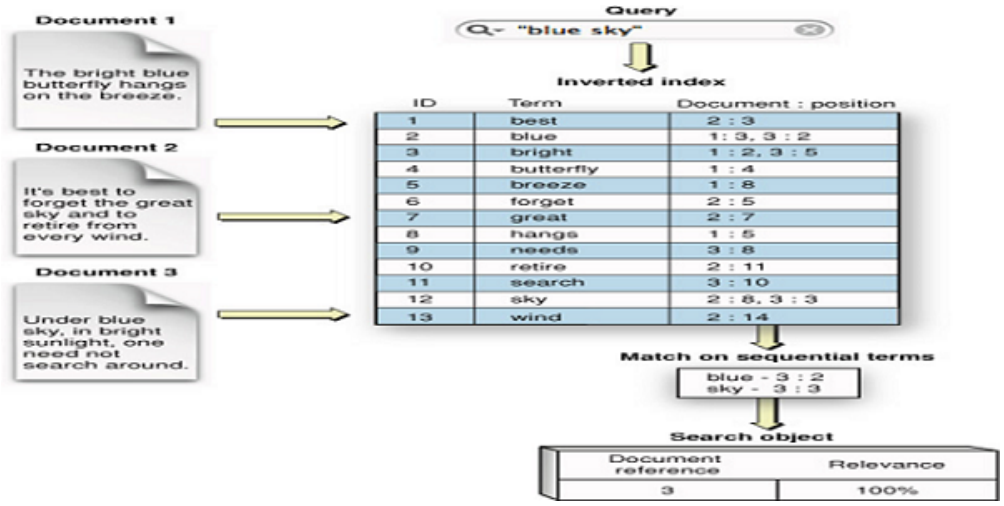

FIGURE 2.3: Flow diagram for Inverted Index

Chapter 3

Experimental Design & Setup

3.1 Tools utilized in this Project

3.1.1 ELK Stack

ELK Stack is combination of Elasticsearch, Logstash, and Kibana used to storing, visualization, and analysis of logs and other time-series data. It is provides end-to-end stack that will be delivers some insights from structured and unstructured data. Elk Stack is open source products, that makes searching, analyzing and visualization of data easier.In ELK Stack:

- Elasticsearch for deep search and data analytics
- Logstash for centralized logging, log enrichment and parsing
- Kibana for powerful and beautiful data visualizations

FIGURE 3.1: Elk Stack Architecture

ElasticSearch

Elastic search is open source search and analytical engine which is capable of solving more number of use cases. It is distributed and RESTful search. Elasticsearch is an Apache Lucene-based search server. In elastic search elastic stack is very important part where data is store centrally to discover the expected and also uncover the unexpected. It is developed in Java programming language and it is used by many big organizations. When you starting with Elasticsearch you should know the basic knowledge of java, JSON, search engine and web technologies. It also run on different platform.The features of Elasticsearch are following-

- Elasticsearch is simple and transparent.
- Elasticsearch is predictable and Reliable.
- Elasticsearch is scalable to large data of Structured and unstructured data.
- Elasticsearch uses to improve the search performance.
- Elastic search is open source and Apache software.
- Elasticsearch is popular search engine, which is using in many organization like Facebook, Wikipedia, eBay, GitHub, stackoverflow, Dell, CISCO, FICO etc.

Logstash

The Logstash event processing pipeline has three stages: inputs filters outputs. Inputs generate events, filters modify them, and outputs ship them elsewhere. Inputs and outputs support codecs that enable you to encode or decode the data as it enters or exits the pipeline without having to use a separate filter.You use inputs to get data into Logstash. Some of the more commonly-used inputs are:

- file: reads from a file on the file system, much like the UNIX command tail -0F
- syslog: listens on the well-known port 514 for syslog messages and parses according to the RFC3164 format
- redis: reads from a redis server, using both redis channels and redis lists. Redis is often used as a ”broker” in a centralized Logstash installation, which queues Logstash events from remote Logstash ”shippers”.
- beats: processes events sent by Filebeat.

Kibana

Kibana 4 is an analytics and visualization platform that builds on Elasticsearch to give you a better understanding of your data. In this tutorial, we will get you started with Kibana, by showing you how to use its interface to filter and visualize log messages gathered by an Elasticsearch ELK stack. We will cover the main interface components, and demonstrate how to create searches, visualizations, and dashboards.The features of Kibana are following-

- Kibana is based on HTML, JavaScript, and Bootstrap
- Kibana is requires a web server, included in the Kibana 4 package, and it is fully compatible with any modern browser
- Kibana is not a requirement for querying the search cluster
- Kibana supports time-based comparisons, easy creation of graphical data representations like plots, charts and maps, flexible and responsive web interface, and a powerful search syntax

3.1.2 Anaconda :Development Environment

It is a distribution of python language for large scale processing, scientific computing , and predictive analytics , that is used for deployment and simplify package management. In Anaconda almost 150 packages are automatically installed.

3.1.3 Spyder

It is the Scientific Python Development Environment It is a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features and a numerical computing environment thanks to the support of IPython (enhanced interactive Python interpreter) and popular Python libraries such as NumPy (linear algebra), SciPy (signal and image processing) or matplotlib (interactive 2D/3D plotting).

3.2 Programming Languages:

3.2.1 JSON (JavaScript Object Notation)

JSON (Javascript object notation) is a lightweight text-based open standard designed for human-readable data interchange. Conventions used by JSON are known to programmers, which include C, C++, Java, Perl, Python etc.

- JSON remains for JavaScript Object Notation.
- The configuration was determined by Douglas Crockford.
- It was intended for human-readable data interchange.
- It is extended from JavaScript scripting language.
- The file extension is .json.

3.2.2 Python

Python has been a very popular programming language for a long time, used by many companies, scientists, casual and professional programmers (apps, cloud/web services, and web sites), and app scripters.

Python has become one of the most popular dynamic programming language along with the other languages such as Perl, Ruby and others. Python and ruby have become especially popular for building the websites using numerous web frameworks like Rails (Ruby) and Django, flask (Python) and Web2Py. Such languages are often called as the scripting languages.

Among the interpreted languages, python is distinguished by its large and scientific computing community. Adoption of python for scientific computing in both the industry applications and academic research has increased.

For data analysis, exploratory computing and data visualization python can be preferred over many other programming languages and tools which are widely in use, such as, R, MATLAB, SAS, and others. Python has also improved its library support for many data manipulation tasks. Combined with pythons strength in general purpose programming, it is a good choice as a single language for building data-centric applications.

Python has many good properties:

- It is high-level, interpreted, object-oriented scripting language supporting the development of the wide range of applications from simple text processing to browsers to games.
- Easy to read and maintain, easy to learn with few keywords, simple structure and a clearly defined structure and syntax.
- It has broad standard library and is portable and cross-platform compatible on UNIX, windows and Macintosh.
- Python supports interactive mode of programming allowing interactive testing and debugging the snippets of code.
- It is an extendable language.
- It provides interface to all major databases.
- It supports GUI applications that can be created and ported to many systems like Windows, Macintosh, and UNIX.
- It is a scalable language providing a better structure and support for large programs then just shell scripting.
- Easily it can be integrated with various languages such as C, C++, CORBA, JAVA.
- It has many other functionality such as automatic garbage collection, support for functional, structural programming methods as well as OOP, dynamic type checking, and dynamic data types Due to the above mentioned features python has been chosen as the development programming language for this system.

3.3 Architecture of Email Analytics

The figure given here is the architectural design of our Email Analytics. In our proposed system, the data is provided to the system. The data taken from Enron development. For the different emails formats the different preprocessing techniques will works.

FIGURE 3.2: Basic Email Analytics Architecture

3.4 Implemented Modules in Email Analytics

Email Dataset

In Enron dataset, the data set size is 2.5 GB and it having 517,440 total email messages. Enron email dataset available at http://www.cs.cmu.edu/ enron/.Enron Dataset having different format of emails like Eml,mbox,pst etc.

Data Preprocessing

In data preprocessing,Email analytics consist of two integrated parts.Elasticsearch is used to create the search indexes for various email fields and extracted entities. The Enron dataset having approximately 517,000 emails.The lucene indexing is efficient and used with other forensic tool as well as Solr and Elasticsearch which is also used for some forensic tools.

ELK stack is a popular tool of the Elasticsearch, Logstash and Kibana. ELK is an end to end stack which can be handles everything from data aggregation to data visualization. I need database with a schema less data models for the purpose of aggregated queries and fast searching. I have two options Elasticsearch and Solr(both are based on Apache Lucene).I decide to go with the elastic search because of the full stack and AWS support. You will understand how to run all three components of elk stack used for analyze data.

Elasticsearch

Elasticsearch is full text search engine.It is provides a REST API over the multiple indexes that can be search and queried.In elk stack indexes are automatically created when you are posting a JSON document to an index scheme.

The index scheme having three parts:

- Index name
- Index type
- Document ID

MAPPINGS:
You need to define schema for the index,before you can run the aggregate queries.So that means Elasticsearch needs to know the data types(integer,String,double) of the attributes in the schema.Elasticsearch does try to guess the attribute type,but also get predictable results with a schema. Snippet from a mapping configuration is following:

```
{
  "mappings": {
    "Product": {
      "properties": {
        "recordType": {
          "type": "string",
        },
        "dateTime": {
          "type": "date",
          "format": "epoch_second"
        },
        "price": {
          "type": "double"
        },
        "quantity": {
          "type": "integer"
        }
      }
    }
  }
}
```

FIGURE 3.3: Mapping In Elasticsearch

Analyzed Fields:

String attributes are analyzing the full text search.This has unwanted side effects when you want to use these kind of fields for aggregation.If an attribute is not needed for full-text-search,

```
"recordType": {
  "type": "string",
  "index": "not_analyzed"
}
```

FIGURE 3.4: Not analyzed In Elasticsearch

Logstash

Logstash is an event of collection and forwarding pipeline.Here number of input,filter and output plug-ins are easy transformation of events.which can specifies minimum an input and output plug-in.

```
input {
  file {
    path => "/data/input.csv"
  }
}

filter {
  csv {
    columns => [
      "recordType",
      "dateTime",
      "productName",
      "productCode",
      "price",
      "qty"
    ]
  }
}

output {
  elasticsearch {
    hosts => ["${ESHOST}"]
    index => "logstash-%{+YYYY-MM}"
  }
}
```

FIGURE 3.5: Creating Logstash configuration

TEMPLATES: Templates are applied when indexes are already created.In raw attributes to be generated when data is added to it,template mapped to index is needed.

Kibana

Kibana is visualization tool for exploratory data analysis. Kibana connect with Elasticsearch node and has access to all kind of indexes on the node. You can select one or more indexes and attributes in the available index for queries and graphs.

DISCOVERY: Open Browser http://localhost:5601/ and Kibana dashboard will open up. You should see the Settings where you need to select at least on index. DISCOVER tab presents the count of records spread over the selected range of time by day. You can adjust this to be less granular (week, month, year) or more granular (hour, minute, second).

FIGURE 3.6: Discovery of time by day

You can use the search bar for terms and page update with results and new graphs.

The ELK stack is used to provide a simple way to analyze data sets. It is not meant to be a statistical analysis tool, but more suited for business intelligence use cases. I found that elasticsearch is very fast while it reads, this is because writes create the indexes on the attributes and the analysis of some attributes as well.if the destination like elasticsearch is not able to keep the velocity, Logstash may drop the data due to back pressure. Elasticsearch is not used as the authoritative data source as it may drop data in case of network partitions.

Visual Analytics

Visualize: Visualize tab open the possible visualizations-the most visualization start with simplest metric-total count of all documents.

FIGURE 3.7: Visualization of Count

DASHBOARD: You can create dashboard form the saved visualizations and save dashboard. You can also set it to auto refresh, which will modify the visuals as new data is collected by Elasticsearch.

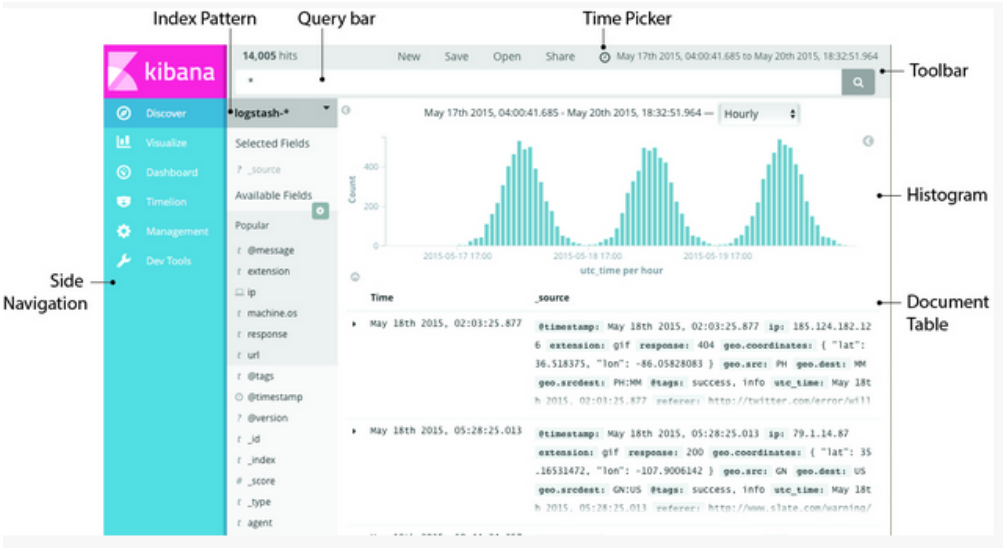

FIGURE 3.8: Dashboard Visualization

Relevant Emails

In the process of analyzing an enron email data set starts with a user defined keyword search that can be filtering for specific senders and receivers.the elasticsearch search engine gives the relevant data based on the search keywords.this gives the investigator greater flexibility to expand and reduce the email dataset.the loop of the search algorithm gives the better results are achieved.

Chapter 4

Results and Implementation work

4.1 Development of the Email Analytics

we represent technique to discovery of evidence and information in investigation from a large email dataset. So in any large email dataset to prevent the investigator form conducting a manual search.

In this book Visualization analysis and Design (Tamara Munzner 2015) Tamara Munzner given explanation about the visual analytics when the exact questions are not known. Email analytics ability to find the human pattern, trends and anomalies.it is very difficult to investigation when content of emails are change.

Elk Stack is most popular open source solutions for not only logs management but also for data analysis. Elk stack is used to effectively and efficiently perform on big data analysis. For analysis take some mailbox data like Gmail, yahoo, Hotmail, rediffmail etc. In general each message. It is present and every message throws lightweight on the communication which individuals square measure having. Every organization want to analyze their corporate mails for trends and patterns. As a reference, I will take my own mails from Gmail account. After downloading the mails from Gmail account we are saving that data into local machine.

As a reference, I will take my own mails from Gmail account. After downloading the mails from Gmail account we are saving that data into local machine.

Downloading mails form Gmail Manually:

- Login to the Gmail - > Go to My Account
- Go to Personal info and privacy

- Then Go to Control your content
- Right side Download your data option - > Click on CREATE ARCHIVE ->
- Then Select data to include only Gmail - > Click on Next - >
- then Customize archive format :File type -> Zip Archive size (max) ->
- upto 50 GB Delivery method -> Add to Drive
- Click on Create Archive -> Download the data and Save in Drive

But here size of the mailbox is every small due to that I will take the well-known Enron email dataset and it has huge collection of Enron emails. That data having Mbox format. We need to again transform Mbox format into a single JSON file.

4.2 Getting data from Enron corpus:

Enron email dataset in a raw form is available at http://www.cs.cmu.edu/ enron/. It is organizes a collection of mailboxes by person and folder. The basic structure of the enron corpus after downloaded and unarchived it. Go ahead with command prompt so that you become familiar with enron email dataset.

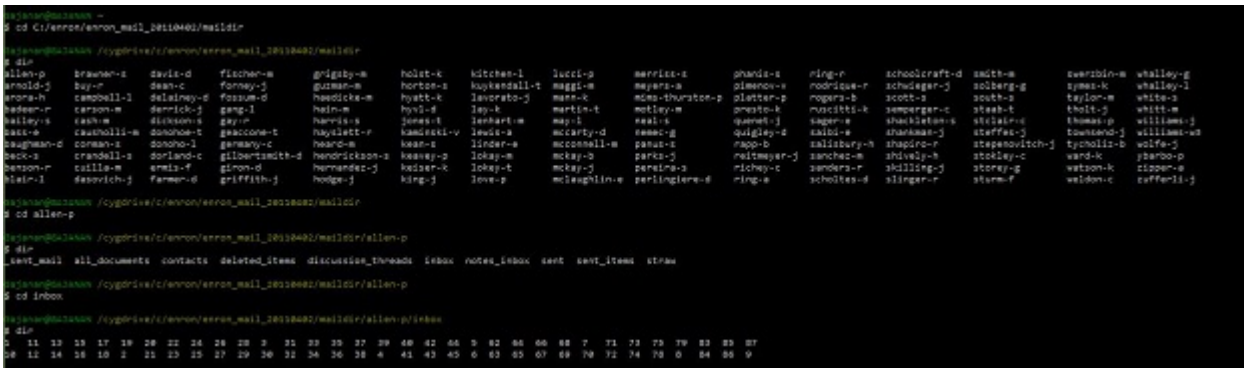

FIGURE 4.1: Familier with email dataset through cmd

FIGURE 4.2: Familier with email format through cmd

Next step is to covert enron mail data into mbox format. An mbox is a large text file of concatenated enron mail messages that are easily accessible by text-based tools. I have used python script to convert enron mbox format. After this mbox file would be converted into JSON format. JSON format is compatible with Elk Stack. A snippet of JSON file is following:

FIGURE 4.3: Json File Format

When you have large amount of data to be pushed into Elasticsearch then it better to do bulk import by using the data file. Each enron mail message is in a line of its own associated with a specifying the index (enron) and emails. In Elasticsearch is automatically specify the id.

Data in the Elasticsearch can be generally classified into two types Exact values and Full text. Exact values are exactly as what it sounds like. Example are a User ID or a date, but also include exact strings such as email address or a username. Whereas full text refers to textual data. First you need to specify the mapping, which can be shown below.

```
DELETE /enron
PUT /enron
{
"settings":
{
    "number_of_shards": 5,
    "number_of_replicas": 1
},
"mappings":
{
    "inbox":
    {
        "_all":
        {
            "enabled": false
        },
        "properties":
        {
            "To":
            {
                "type": "string",
                "index": "not_analyzed"
            },
            "From":
            {
                "type": "string",
                "index": "not_analyzed"
            },
            "CC":
            {
                "type": "string",
                "index": "not_analyzed"
            },
            "BCC":
            {
                "type": "string",
                "index": "not_analyzed"
            }
        }
    }
  }

}
```

FIGURE 4.4: Enron mapping

You can verify that the mapping has indeed been set.

```
curl -XGET "http://localhost:9200/_mapping?pretty"
{
    "enron" :
    {
        "mappings" :
        {
            "inbox" :
            {
                "_all" :
                {
                    "enabled" : false
                },
                "properties" :
                {
                    "BCC" :
                    {
                        "type" : "string",
                        "index" : "not_analyzed"
                    },
                    "CC" :
                    {
                        "type" : "string",
                        "index" : "not_analyzed"
                    },
                    "From" :
                    {
                        "type" : "string",
                        "index" : "not_analyzed"
                    },
                    "To" :
                    {
                        "type" : "string",
                        "index" : "not_analyzed"
                    }
                }
            }
        }
    }
}
```

FIGURE 4.5: Verify mapping

Now lets load all the mailbox data by using the json file, in the following manner:

```
curl -XPOST "http://localhost:9200/_bulk" --data-binary @enron.json
```

FIGURE 4.6: Load Mailbox Data Using josn File

We can check if all the data has been uploaded successfully.

```
curl "localhost:9200/enron/inbox/_count?pretty"
{
    "count" : 41299,
    "_shards" :
    {
        "total" : 5,
        "successful" : 5,
        "failed" : 0
    }
}
```

FIGURE 4.7: Data uploaded successfully

You can see total 41299 records having different messages, have been uploaded. Now lets start with some analysis on this data. Kibana provides very good analytic capability and associated charts.

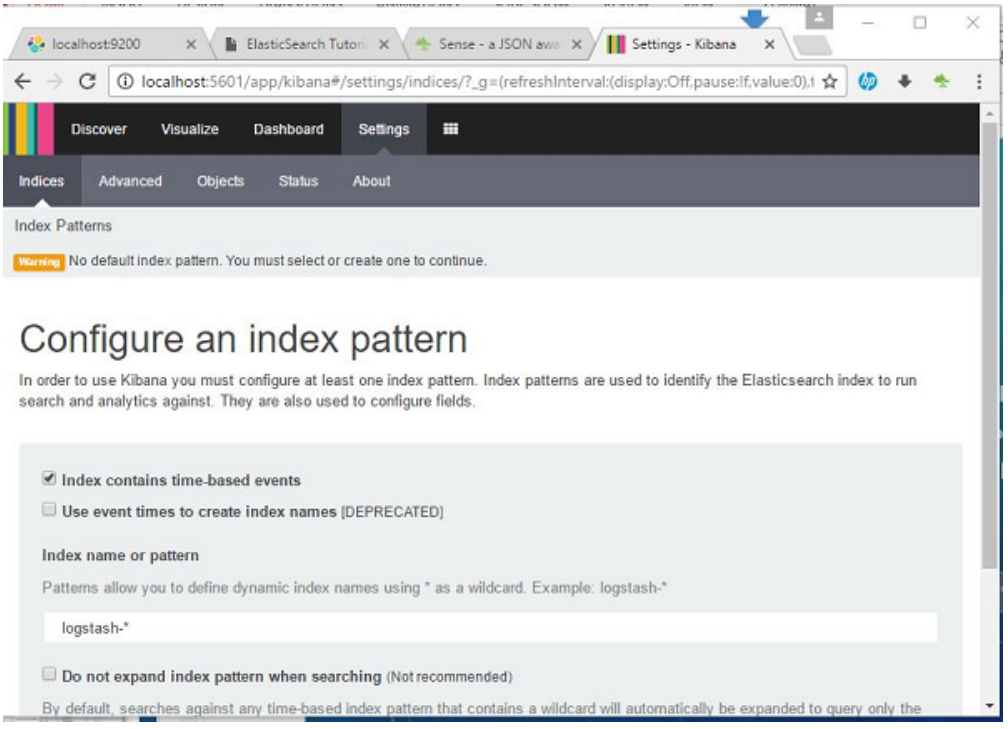

FIGURE 4.8: Configure an index pattern

The above diagram shown configuration of the Kibana . In that you can create different indexes in Kibana.An index pattern identifies one or more Elasticsearch indexes that you want to see the exploration with Kibana. Kibana looks for index names that match the specified pattern

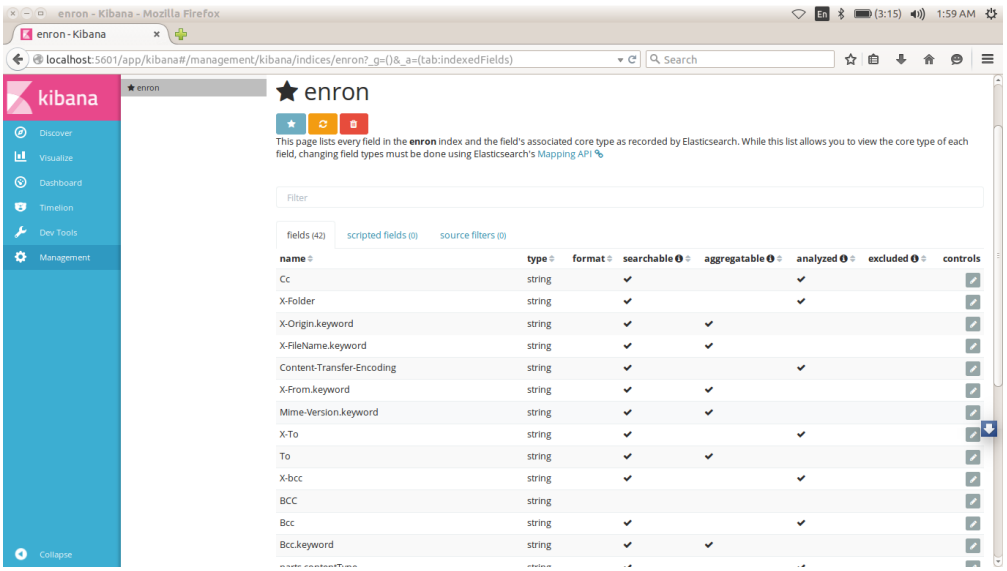

FIGURE 4.9: Enron Index

Setting the default Index pattern: In Kibana, the default index is loaded by automatically when you click on Discover tab. Kibana when you press on the star to the left of patterns list of the setting. The default index pattern:

- Goto settings ¿ Indices.
- Select default pattern you want in index pattern list.
- Click on Favorite button.

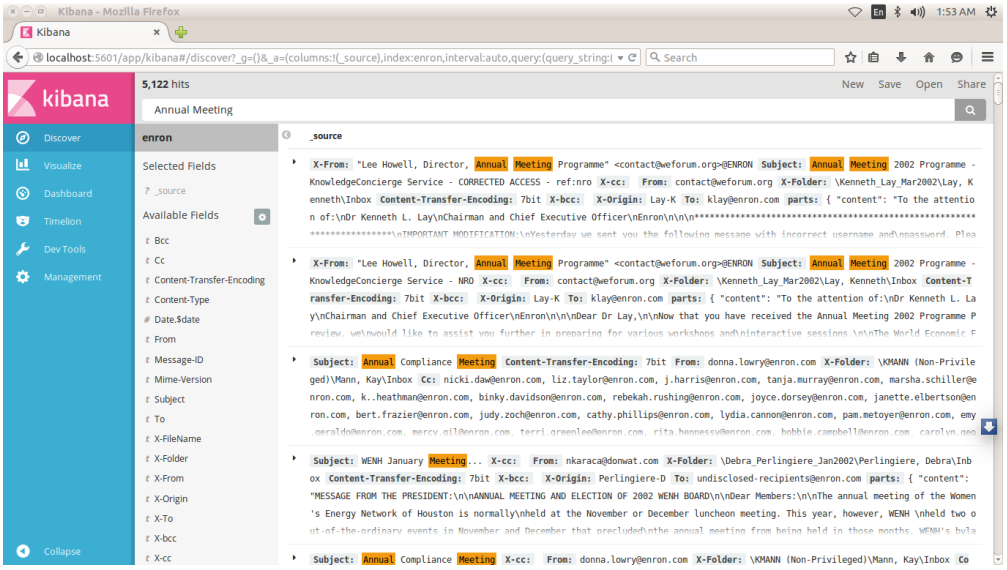

FIGURE 4.10: Enron Searching

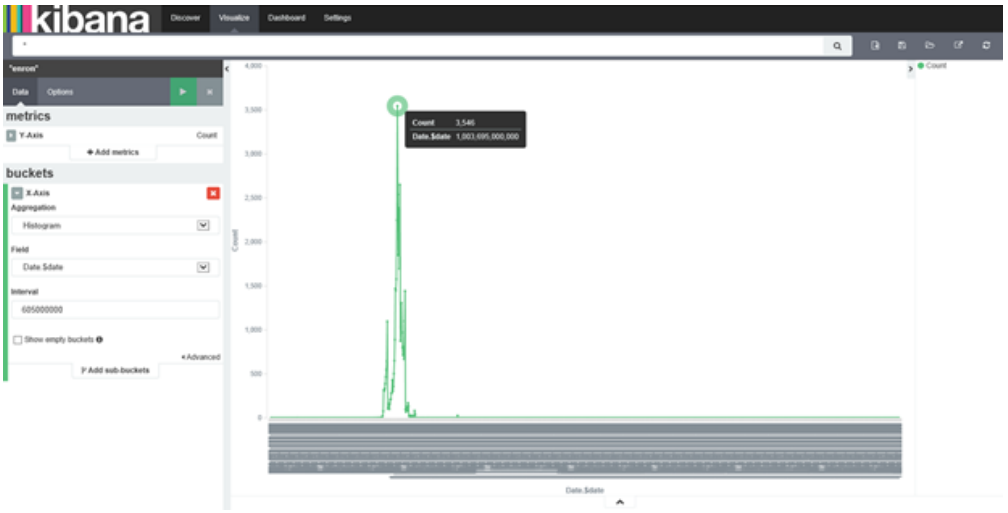

FIGURE 4.11: Enron histogram on a weekly basis

The above histogram shows the messages which spreads on a weekly basis. The date value is in terms of milliseconds. You can see that one particular week has a peak of 3546 messages. There must be something interesting happening that week. Now lets see who the top recipients of messages are.

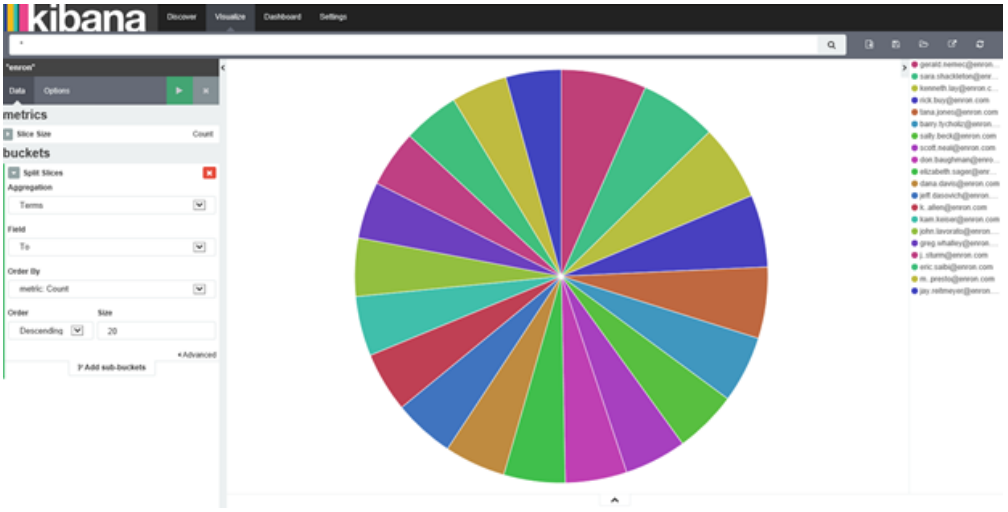

FIGURE 4.12: histogram shows the messages which spreads on a weekly basis

In the above pie chart, you can see that Gerald, Sara, Kenneth are some of the top recipients of messages. Now lets check out the top senders?

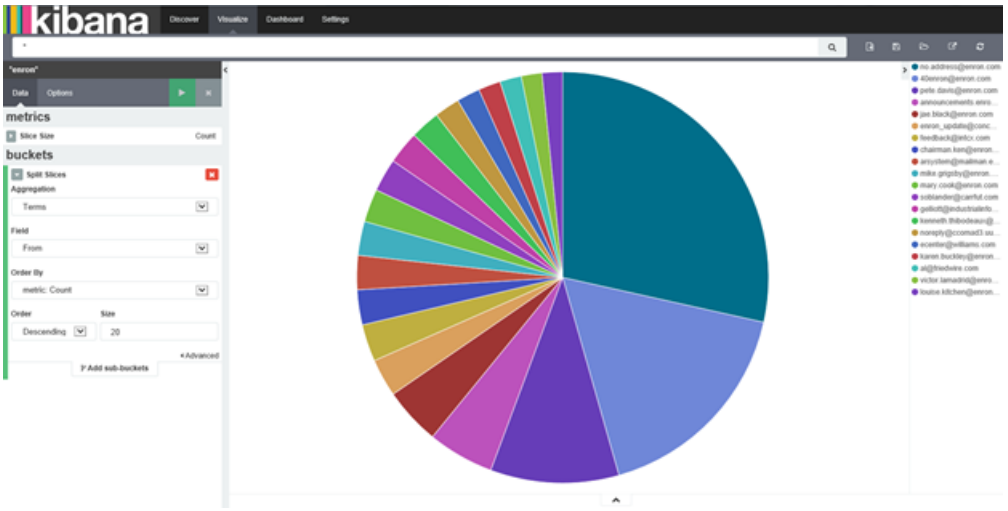

FIGURE 4.13: top recipients of messages

We can see that Pete, Jae and Ken are the top senders of messages. In case you may be wondering what exactly Enron employees used to discuss, lets check out top keywords from message subjects.

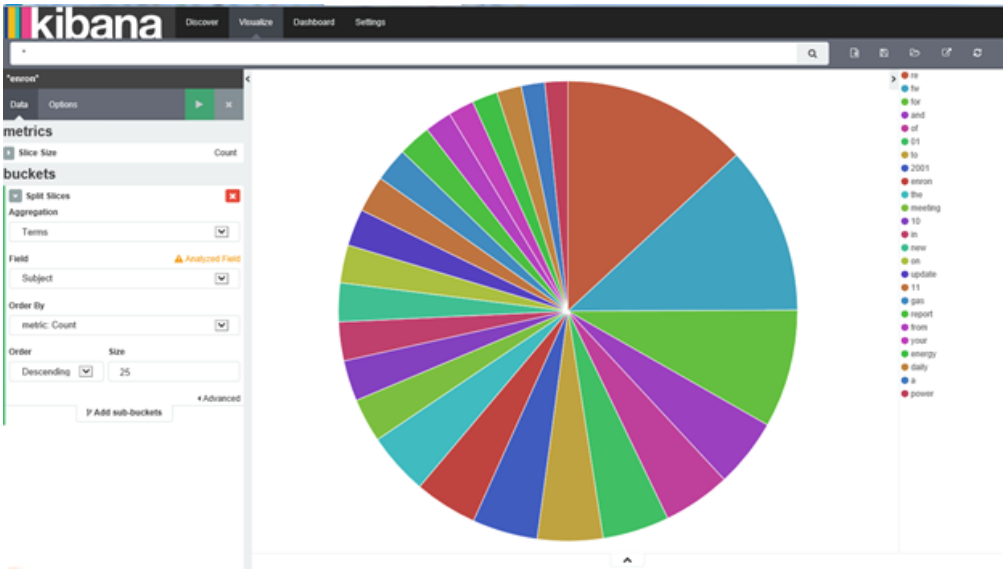

FIGURE 4.14: top keywords from message subjects

It seems that most interesting discussions focused on enron, gas, energy, power. There a lot more interesting analysis can be done with the Enron mail data.

Enron data can be visualized using different faces found across within each email and multiple email archives. Kibana indexes provides full text search feature and visualization for better decision making.

Create a kibana visualization dashboard to view all senders and receivers of enron email within any archive.

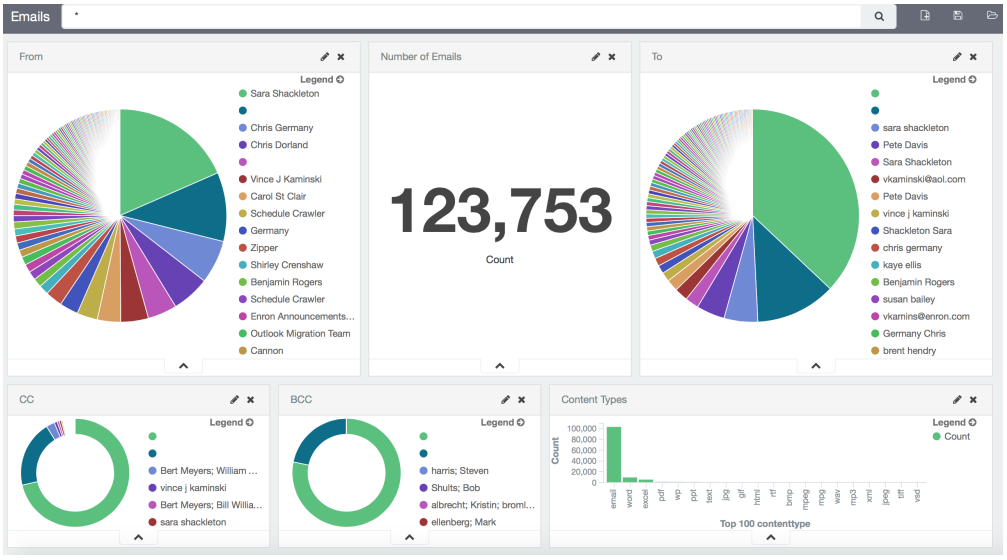

FIGURE 4.15: Enron Visualization 1

Type in kibana search box for visualization and view result.

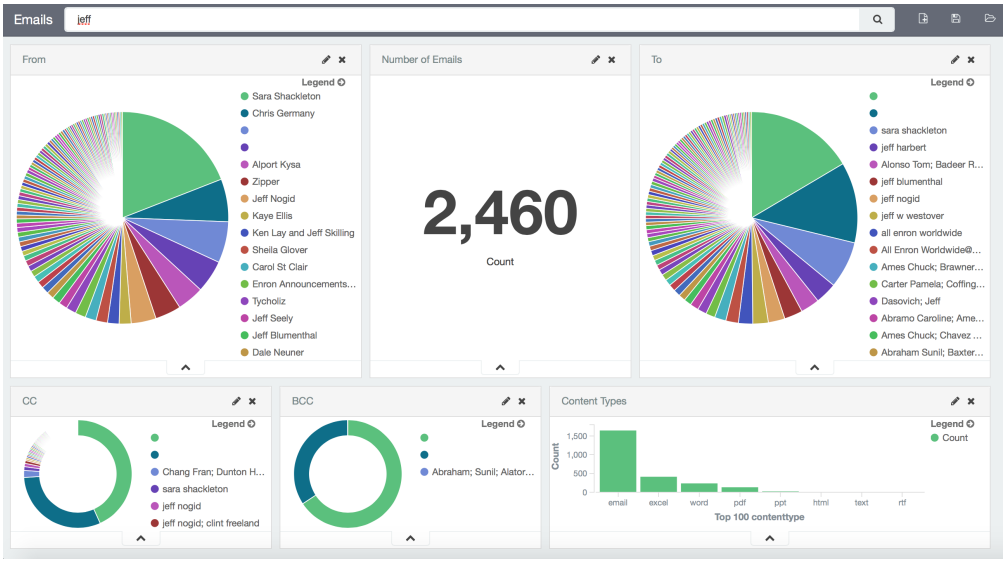

FIGURE 4.16: Enron Visualization 2

Simply click on any value within any chart of Kibana to filter down the results and the visualization.

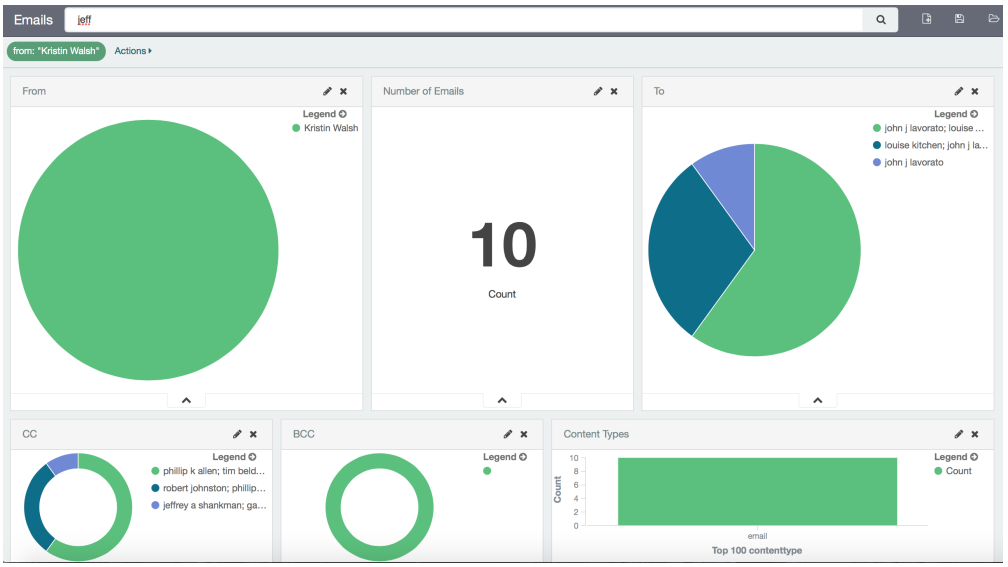

FIGURE 4.17: Enron Visualization 3

Python Code:

```
1
2   import re
3   import email
4   from time import asctime
5   import os
6   import sys
7   from dateutil.parser import parse  # easy_install python_dateutil
8
9   directory = sys.argv[1]
10
11  for (root, dirs, file_names) in os.walk(directory):
12
13      # if root.split(os.sep)[-1].lower() != 'inbox':
14      #     continue
15      for file_name in file_names:
16          if file_name == "DELETIONS.txt":
17              continue
18          file_path = os.path.join(root, file_name)
19          # print "Processing ", file_path
20          message_text = open(file_path).read()
21
22      # compute fields for the From_ line in a traditional mbox message
23
24          _from = re.search(r"From: ([^\r]+)", message_text).groups()[0]
25          _date = re.search(r"Date: ([^\r]+)", message_text).groups()[0]
26
27      # convert _date to the asctime representation for the From_ line
28
29          _date = asctime(parse(_date).timetuple())
30
31          msg = email.message_from_string(message_text)
32          msg.set_unixfrom('From %s %s' % (_from, _date))
33          print(msg.as_string(unixfrom=True))
34          print
35          # redirect stdout to a file, or write to a file directly
36
```

FIGURE 4.18: mailboxes convert enron inbox to mbox

```
import sys
import mailbox
import email
import quopri
import re
from bs4 import BeautifulSoup

try:
    import jsonlib2 as json  # much faster then Python 2.6.x's stdlib
except ImportError:
    import json

MBOX = sys.argv[1]
OUT_FILE = None
try:
    OUT_FILE = sys.argv[2]
except Exception, e:
    pass

def cleanContent(msg):

    # Decode message from "quoted printable" format

    msg = quopri.decodestring(msg)

    # Strip out HTML tags, if any are present

    soup = BeautifulSoup(msg, "lxml")
    return ''.join(soup.findAll(text=True))

def cleanDate(date):
    return re.sub(r' \(...\)', "", date)

def jsonifyMessage(msg):
    json_msg = {'parts': []}
    for (k, v) in msg.items():
        json_msg[k] = v.decode('utf-8', 'ignore')

    # The To, CC, and Bcc fields, if present, could have multiple items
    # Note that not all of these fields are necessarily defined

    for k in ['To', 'Cc', 'Bcc']:
        if not json_msg.get(k):
            continue
        json_msg[k] = json_msg[k].replace('\n', '').replace('\t', '').replace('\r'
                , '').replace(' ', '').decode('utf-8', 'ignore').split(',')

    try:
        for part in msg.walk():
            json_part = {}
            if part.get_content_maintype() == 'multipart':
                continue
            json_part['contentType'] = part.get_content_type()
            content = part.get_payload(decode=False).decode('utf-8', 'ignore')
            json_part['content'] = cleanContent(content)

            json_msg['parts'].append(json_part)
            json_msg['Date'] = cleanDate(json_msg['Date'])
    except Exception, e:
        sys.stderr.write('Skipping message - error encountered (%s)' % (str(e), ))
    finally:
        return json_msg

#Note: opening in binary mode is recommended

mbox = mailbox.UnixMailbox(open(MBOX, 'rb'), email.message_from_file)
def gen_json_msgs(m_box):
    while 1:
        msg = m_box.next()
        if msg is None:
            break
        print(json.dumps(jsonifyMessage(msg)))

gen_json_msgs(mbox)
```

FIGURE 4.19: mailboxes jsonify mbox

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Here I have introduced a new era of analytics that is E-mail Analytics by proposing a new methodology of searching and mining of useful E-mails from large email datasets. Analytics over email dataset makes easier to investigate for investigators to identify hidden relationships and anomalies within the email datasets. This will improve and speed up the results of the investigation process. To find relevant emails, entities and correspondents in the email data sets, investigators found new and interactive technique of visuals which helps them in their decision making. Once these emails are found relevant which were hidden from the initial search through our search result reduction techniques can be brought back into final detailed examination. My study and approach demonstrated visual interaction which can reduce the size of results set, so that the remaining emails can be examined in detail to find relevant emails through investigation.

5.2 Future work

The agenda is analyzing the topic modeling using the elk stack.E Mails and other text forms of communication such as tweets and text messages present a unique challenge due to their nature of the communication.we are not aware of any search system and indexing that have implemented an efficient methods for creating a new index for a subset of document set from index of full document set

Bibliography

[1] Bernard Kerr. Thread arcs: An email thread visualization. IEEE Symposium on Information Visualization, 2003.

[2] C.Ramasubramanian and R.Ramya. Invest: Intelligent visual email search and triage,dfrws usa 2016-proceedings of the 16th annual usa digital forensics research conference, digital investigation. DFRWS USA 2016, 18, 2016.

[3] John Haggerty, Sheryllynne Haggerty, and Mark Taylor. Forensic triage of email network narratives through visualisation. Information Management and Computer Security, 22, 2014.

[4] John Haggerty, Sheryllynne Haggerty, and Mark Taylor. Enron corpus dataset. Information Management and Computer Security, https://www.cs.cmu.edu/ ./enron/.

[5] Haggerty J, Karran AJ, Lamb DJ, and Taylor M. A framework for the forensic investigation of unstructured email relationship data. International Journal Digital Crime Forensics, 2011.

[6] https://lucene.apache.org/.

[7] https://lucene.apache.org/solr.

[8] https://www.elastic.co/.

[9] Enron Dataset,http://www.cs.cmu.edu/ enron/.

[10] Maguire E. ,Munzner T. Visualization analysis and design. AK Peters visualization series. Boca Raton, FL- CRC Press; 2015.