Alma Mater Studiorum · University of Bologna

School of Science
Master Degree in Computer Science

A Visual Framework for Graph and Text Analytics in Email Investigation

Supervisor:
Professor.
Danilo Montesi

Candidate:
Ivan Heibi

Session I
Academic year 2016/2017

Abstract

The aim of this work is to build a framework which can benefit from data analysis techniques to explore and mine important information stored in an email collection archive. The analysis of email data could be accomplished from different perspectives, we mainly focused our approach on two different aspects: social behaviors and the textual content of the emails body. We will present a review on the past techniques and features adopted to handle this type of analysis, and evaluate them in real tools. This background will motivate our choices and proposed approach, and help us build a final visual framework which can analyze and show social graph networks along with other data visualization elements that assist users in understanding and dynamically elaborating the email data uploaded. We will present the architecture and logical structure of the framework, and show the flexibility nature of the system for future integrations and improvements. The functional aspects of our approach will be tested using the enron dataset, and by applying real key actors involved in the enron case scandal.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Users use emails to deal with a lot of daily life situations, whether they refer to personal matters or to work duties. A better understanding of this phenomena and the context itself of messages we send and receive can help us build the profile of who we really are, and what type of relations we have with our contacts. Such approach could be summarized in a famous quote from Johann Wolfgang von Goethe: "Tell me with whom you associate, and I will tell you who you are". The success of such analysis is related to the way we are going to model the data, the techniques adopted to elaborate it, what are the analysis to execute, and how we will represent the final results and visualize them.

1.2 Motivation and objectives

The traditional email clients store all our messages content with their related header metadata (sender, receiver, subject etc), so they represents the basic and primary platform model where users can perform traditional operations like a generic keyword search or configure their contacts list. However some unconventional analysis, e.g: examining the number of messages

exchanged between two addresses from the contacts list, is a hard job to accomplish. If we consider the fact that we might also have the necessity to deal with large archives and messages quantity, these operations will turn out to be really expensive to be done manually by a human. Using automated elaboration techniques, can facilitate these operations, and taking advantage of such new methods of analysis can infer new additional hidden information, specially from the elaboration of a big dataset.

Building a tool that can deeply analyze and mine a large amount of messages/data, and create a friendly visualization of the information and results, will turn out to be very beneficial. Such tool could be potentially used as a mechanism to analyze the email account of people implicated in juridical issues, or it could help single users to detect anomalies and classify their contacts according to the messages they exchange, by viewing their email data from different perspectives.

Our final object is to create a framework which implements basic features already diffused in similar systems with the adoption of new useful modifications and improvements, along with the presentation of new features that can benefit from innovative data elaboration techniques and textual mining methods adaptable to email data elaboration.

1.3 Research process

Since the final object is creating a usable and functional framework, achieving this result comes with the combination of different techniques and fields of study: text mining, forensic analysis, data elaboration, visual data representation... etc. Once we have a clear idea of what our system needs, we need to integrate all these parts correctly in one final container.

All the related works analyzed associated to the various fields of study mentioned above, treat some basic concepts, and try to apply innovative techniques to handle past and new questions. In our approach we will treat individually each material and point out the most relevant aspects that might

become useful to realize our final system.

To get the actual real effect of applying all the variety of these techniques, we will analyze numerous commercial email forensic tools, test their usability, and compare them according to important and common features. The final results will help us understand where should we focus on when building our framework.

1.4 Roadmap

This work will start with some basic theoretical background definitions of Natural language processing methods, along with the definition of the most common formats of email data. After that we will cover all the relevant studies in literature in relation to the study fields already mentioned: forensic and data analysis, and graphical representation of the data.

The 3rd paragraph will talk about the logical and conceptual decisions made in the definition of the framework theoretical basis. This involves: (a) The data pre-processing elaborations, e.g: data cleaning, data conversation. (b) The email mining operations, e.g: community detection, concept classification. The features included in the framework are not final, therefore we will dedicate a section inside this paragraph to talk about possible future modification and applications for each previous field.

The 4th paragraph will emphasize the attention on the implementation aspects, and the architectural structure of the application. So we will cover the two phases: initialization and run-time of the framework. In addition, we will present the basic graphical components, and how to interact with them.

The final paragraph will evaluate our framework based on his basic features discussed in the 3rd and 4th paragraph. The evaluation will have effect in two steps: (1) Testing the validation of the main features through randomly selected sub-data from the Enron dataset, (2) Applying a case of study "The Enron scandal" as a matter of investigation, and infer relevant forensic information/clues related to the real facts and juridical reports. Finally, we

will give our final conclusions and thoughts on the work.

Chapter 2

Background and state of the art

The analysis of email collections can be done with several approaches and inferring different results depending on what we want to observe. Since we want to produce a final usable framework which incorporates different features, several fields of study should be taken in consideration, and we need to find the most suitable way to mix them in a one integrated system, which can handle some forensic examinations requests, and correctly generate and visually emphasize the most significant information.

We took in consideration some important macro fields of study: text mining, forensic analysis, data elaboration, data mining, and visual data representation. Our approach will examine the most relevant works covered in literature related to these fields of study, and review the possibility of merging them for the realization of an integrated final tool. We will point out the basic, yet fundamental concepts, along with new innovative techniques for handling past and new issues. Later in this chapter we will get deeper and treat separately these fields.

Many of the topics and techniques studied in literature are already deployed in numerous commercial email forensic tools. We will make a brief inspection of these tools pointing out the most common features included, and make a conclusive comparison between them. This will help us answer the questions: 'what an email forensic tool should do?', and 'what features

should be reinforced?'. In addition we will try to add new possible features based on our state of the art overview, and the necessities arose due to the information we want to obtain.

Before we present the related works in literature for the study fields of interest we just mentioned, we would like to give a brief definition/background on some Natural Language Processing (NLP) techniques. The aim of NLP procedures is to process human language with automatic or semi-automatic techniques, these techniques are very useful if applied on the textual content of emails. So our first section, will be a generic background for the most popular and useful methods we should take in consideration, this will help us better understand the features treated later.

2.1 Background on NLP

Textual evidence is generally a very important part of a forensic investigation. Mining correctly the text and presenting the searching hits properly enables the investigator to find relevant and hidden semantic meanings. Text mining can refer to different fields like: information extraction, topic tracking, content summarization, text categorization/classification and text clustering.

We will present some popular and interesting algorithms for two important text mining sub-fields: text weighting and text classification. Later we will see how these techniques could be applied and integrated to help us in the email data analysis applications.

2.1.1 Text weighting and indexing

Text or term weighting is the job of assigning the weight for each term (word), in order to measure the importance of a term in a document. A very important and popular tool used in natural language applications is the Term Frequency Inverse Document Frequency (TFIDF): it's a statistical weighting scheme, which determines the relative frequency of words in a

specific document compared to the inverse proportion of that word over the entire document corpus. This method will determine the relevance of words in particular documents, so words that tend to appear in a small set of documents will have a higher TFIDF value [29]. More formally, given a document collection D and a word w, we can can calculate the TFIDF value as: $tfidf(w, d, D) = tf(w, d) * idf(w, D)$ where $tf(w, d)$ is the frequency number of word w inside document d, and $idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$ with N total number of documents, and on denominator we calculate the number of documents where the word w appears.

From the mathematical definition we can notice that high frequency words that appear in a lot documents will not obtain a high score, and therefore appear less relevant (e.g: 'the'), some of these words with very low TFIDF score are included in a set of words called 'stop words'. Stop words are terms which have very little meaning (e.g: 'the','and','is'...etc), they get filtered and removed before weighting the text, in the data preprocessing and preparation phase. This set of words is different according to the textual language processed.

The vocabulary of words used in TFIDF must also include meaningful word phrases (combination of several words) and not only the single words. A very popular technique used to build this vocabulary and generate word phrases is the n-gram model. This operation is also called text indexing, the main object of the n-gram model is to predict a word w_i based on the previous n words: $w_{i-(n-1))}...w_{i-1}$. For instance if a word probability depends only on the previous word we call the model bigram, in case it's conditioned to the previous 2 words then we call it a trigram model. Another notable text indexing methods is the ontology-based approach: a formal declarative definition which includes vocabulary for referring to terms in specific subjects areas along with logical statements which can describe the relationships between the words.

Vector space representation of documents and queries using the above indexing and term weighting techniques enjoys a number of advantages in-

cluding the uniform treatment of queries and documents as vectors. However, an interesting problem arising is the inability to cope with two classic natural language problems: synonymy and polysemy. Synonymy refers to a case where two different terms have the same meaning, while a polysemic term is a term with more than one meaning. Next we will introduce a very interesting method of text classification that might handle these kind of problems.

2.1.2 Text classification

For the email case, the task of text classification, which includes information retrieval (IR) and text categorization is very helpful for our needs, this task is mainly concerned with two kind of properties from the indexing term: semantic quality and statistical quality [32].

A very interesting and high performing algorithm used to solve these kind of problems is LSA (Latent Semantic Analysis) [27]: an indexing method that uses truncated SVD (Singular Value Decomposition) technique to decompose the original matrix of words frequencies in documents. Sometimes when applied in information retrieval context, it is also called Latent Semantic Indexing (LSI).

SVD is a matrix decomposition method that decomposes original matrix to left singular vectors, right singular vectors and singular vectors, formally the SVD of a matrix X is: $X = U\Sigma V^T$. So if we have X matrix of words/doc occurrences, we can define the correlation between the words like the matrix product XX^T and the documents correlations like X^TX, we can use SVD to decompose these representations, the result will be : $XX^T = U\Sigma^2U^T$ and $X^TX = V\Sigma^2V^T$, these final definitions shows us that U must contain the eigenvectors of XX^T, while V must contain the eigenvectors of X^TX, if we apply this to our original matrix X we will get this representation :

$$X = \begin{bmatrix} & & \\ u_1 & \dots & u_l \\ & & \end{bmatrix}^U \cdot \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_l \end{bmatrix}^\Sigma \cdot \begin{bmatrix} v_1 \\ \vdots \\ v_l \end{bmatrix}^{V^T} \tag{2.1}$$

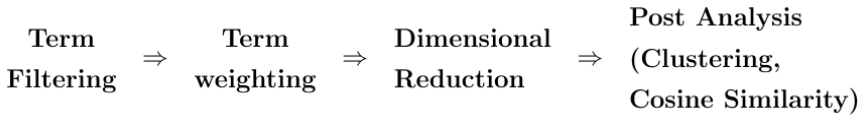


Figure 2.1: Analytic tasks performed in LSA process, from [14]

Task	Methodological choices
Term filtering	+ Frequency-based stoplist + Manually selected stoplist
Term weighting	+ Mostly TFIDF or log-entropy
Dimensional reduction	+ SVD (Singular Value Decomposition)
Post-LSA analysis	+ Cosine similarities, + Classification + Clustering

Table 2.1: LSA analytic tasks, and the corresponding methodological choices

Σ is the diagonal matrix containing the singular values, while the the columns u_i and rows v_i are the right singular values. This representation will help generate k clusters (concepts) and build the corresponding singular vectors from U and V. The new approximation let us have a variety of new operations and combinations with the vectors representation, typically these kind of operations will use the 'cosine similarity' to calculate the closeness and to compare the vectors.

An interesting comparison of text representation techniques in [40] showed that LSA has very high performances in text categorization also when applied on different languages datasets. In addition LSA produced admirable results in documents discrimination and indexing, for both semantic and statistical quality.

The scheme in Figure.2.1 summarizes the analytic phases of the LSA process. Along with this scheme, in Table.2.1 we list the possible methods which can be used for each one of these tasks.

As we can see the second step is 'Term weighting', which we already

talked about in the previous section, the method picked to handle this task has an important impact on the final result produced by LSA. Finding the optimal weighting method for transforming the term frequencies is also widely addressed in the information retrieval literature.

Two previous works [14] and [11] studied the possible term weighting methodologies and their application in the LSA process. Two major term-weighting methods were analyzed: TFIDF and log-entropy. Some experiment comparison results proved that TFIDF appears to be better at discovering patterns in the "core" of the language, so it identifies larger groups of terms which tend to appear all together in a much moderate frequencies, which makes it an appropriate solution when our intent is to represent documents in a relatively conceptual and complex semantic space.

An actual application of TFIDF as pre-LSA term weighting method has been made in [11], the final results obtained were very significant and showed the high efficiency of this applications. Using TFIDF before a matrix decomposition process, was successfully included also in [28], the proposed approach of this study was to classify documents according to their genres by automatically extracting the features (word phrases). The results obtained were very encouraging, a high accuracy of 81.81%, 80% of precision, and 81% of recall, which demonstrates that this approach can contribute positively in solving textual categorization problems.

2.2 Email data structure background

Email nature is complicated, and therefore in order to perform mining and analysis operations on its data we need first to understand the information that they carry. The email structure is basically divided in two parts: the header (metadata), the body (all the context of the email which might include also attachments). The message header contains a lot of crucial information, such as the sender, the receiver and the time when the email was generated. In Table.2.2 we present a list of the fields usually contained in the header

Field	Definition
Message-ID	An automatically generated id consists of timestamp information along with sender account info
Date	The time when the email was generated
From	The sender
To	The recipient
Subject	The email subject
Mime-Version	The Multipurpose Internet Mail Extensions verision
Content-Type	Indicates the presentation style
Content-Transfer-Encoding	The type of transformation that has been used in order to represent the body

Table 2.2: Email header fields and their meaning

and a short description of their meaning.

The message content part might be composed from different elements. A message content type might be: plain text, html content, or multipart. Along with the type of content, the character encoding code should be mentioned, this will permit the correct conversion of the data when reading it. Composing a message as HTML, can still make available the option of sending it as plain text, HTML, or both. In the next sections we will present the fields of study, which will address one or both these email parts to accomplish some specific analysis.

2.3 Email mining and forensic analysis

We can consider Email mining as a sub field of the more generic process of data mining elaboration, the aim is to explore and analyze a large collection of emails in order to discover valid, potentially useful, and understandable

patters behind the data. Since we would also like to associate the results obtained as important information to use for juridical cases, we need to perform these analysis by taking in consideration their relevancy from a forensic point of view.

Email related illegal usage and crime problems have become increasingly prominent. For instance email could be used for spam, spread pornography fraud and other similar negative activities, as a result email has become a potential carrier of criminal evidence for solving cases and providing evidence in a law court.

In this section we will treat individually some interesting email mining threads and present the way they are being handled in literature. Mainly we will focus on: the social network analysis, the spam and contacts identification, and the categorization of the messages .

2.3.1 Social network analysis

The social network analysis (SNA) is an approach to study the human social interactions and dynamics. So such analysis is used to infer community structure and organization patterns between different social actors. If we take a look at the behaviors manifested when using emails, we can see some similarities with the basic characteristics of a typical social network:

Different actors: contacts and email addresses.

Contacts have the ability to connect with each other: we send and receive emails, this way we might create a 2-way communication

Sharing information: we can share different type of data when using emails (e.g: textual, images, attachments etc)

The construction of a social network from a collection of emails, is a very interesting form of representation which will open the doors for new analysis and mining operations. This new representation perspective for emails can

help investigators also view much more easily the social patterns, and the type/strength of correlation between the contacts.

A lot of works have been done to emphasize such aspect, and the growth popularity of social networks platforms might be a big influence for that, although in this case we can call such networks as 'performed': users have a full control on choosing their connections. On the other hand for our case emails hides unplanned communication patterns and contacts communities. This fact leads us to elaborate alternative approaches to build a community network, the common approach of almost all studies is to build the relations without analyzing the actual content of the emails. The success of this operation is closely related to an important pre-processing data phase. Particularly, we must apply the right transformation for the data we have in a comprehensive format for social network analysis.

Most of the studies focus on the messages header to build this social network representation, but they have to define and readapt accordingly the header data in a suitable form. A large number of these previous studies, as also mentioned in the email mining summary made by Guanting Tang et.al[33], used the From, To, and CC fields, as a way to define the links between different actors (email addresses). The studies that adopted this solution, treated different email addresses as unique users. Others pre-elaborate the email addresses and tokenized them before using them as entities.

Practical adaption of these model have been done by several works, for instance by Bengfort et al.[7], who made a further analysis also on how contacts communities should appear and be structured, or by the Immersion tool [23]. The most common graphical representation of this information is through graph networks, further we will get deeper and talk about it, when treating the data graphical visualization techniques.

2.3.2 Email spam and contacts identification

A lot of studies have been done in order to classify emails received from trusted personal contacts and those considered spam. Spam email was always

a major problem for society, this is due to the massive data received along with the fact that in some cases they issue cyber crimes threads (e.g: trick login to phishing sites that can steal personal data).

A lot of works like [38] uses some data mining techniques along with clustering models based on important features common to spam messages. This approach tries to extract as many attributes from the emails as possible, for instance: message id, sender IP address, sender email, subject etc. After retrieving all the attributes needed, clustering operations will try group emails that share same attributes (e.g: the subject), this operation will be repeated for a number I of iterations, and for each iteration $i \in I$ will try to create new clusters from the resulting clusters created on the previous iteration $i - 1$.

Machine learning can also give a good contribution to this detection, through an automated adaptive approach. This can be done with a Text-based processing, which tokenize and extract a bag of words (BoW), also known as the vector-space model, from the messages, and try to represent words in different categories according to their occurrences in the text. This helps us check words that occur often in spam emails and consider such incoming emails as probably spam, this approach follows a Naive Bayes classifier method.

Another common ML method used is the K-nearest neighbor (K-NN) classifier method, in this case the emails are compared, as such when a new email needs to be categorized, the k most similar documents are considered and if the majority belong to a certain category, the new email will also be assigned to same class.

The natural language processing techniques previously mentioned in section.2.1 could be adopted also for spam detection. In fact a text classification and topic extraction can reveal anomalies and rare textual clusters. For instance someone can note a cluster containing textual anomalies related to commercial adds. Works like [5] and [31] showed the applicability and the positive effects of these techniques on spam classification. An additional advantage of using these methods is the fact that they could be applied to

different context languages since words weighting techniques like TFIDF do not take in consideration the semantic definitions, instead analyze the context and the occurrences of the words.

Sender and receiver authenticity and integrity is considered a more general case of spam emails. Many spam emails contain a fake 'From' in the header, so the sender's email address does not really exist, [9] talk about the importance of having the 'Received' field, that can mention the list of all the email servers through which the message traveled before reaching his final destination, a good analysis of this field can create and track the actual path a message has done.

In [8] the contact authentication is combined with message features analysis, two classifiers are used in this case: a spam detection based on messages features, and a further secondary spam determination based on a sub set of the original features for sender determination, if both categorize the message as spam then the system outputs 'spam content'.

2.3.3 Email categorization

By email categorization, we mean the process of creating different groups according to some conditions, and associate the emails to their corresponding category. A lot of users perform this operation manually for each email, by dragging it in the corresponding category. We can consider the spam detection a more specific problem of categorization. The email content is the main part analyzed for this purpose. Elaborating the context and the usage of textual mining methods along with automatic elaboration techniques, can help us accomplish these results.

K-Means algorithm: is a very popular supervised machine learning algorithm to partitioning a set of inputs in k clusters following a defined cost function. In the work of Dechechi et.al[12], we have a good example of this application. They defined a set of documents $D = D_1, .., D_2$ (each one representing a different email), a similarity measure, and the cost function to define the portioning. Giving the number of clusters

k the goal is to compute a membership function $\Phi : D \rightarrow \{0...k\}$, such that it minimizes the cost function and respect the similarity measure between the documents (the distance).

TFIDF classifiers: the idea is to calculate the similarity between categories already discovered and uncategorized documents, so the new document (email) will be assigned to the nearest category taking in consideration the cost function. TFIDF (see Section. 2.1.1 for a background definition) will represent the similarity measure. Each email is defined as word vector, while each different category will be defined by a centroid vector, the centroid will be weighted according to the TFIDF scheme. The similarity between two word-vectors will be calculated according to the 'cosine similarity' principal. Several studies used this type of classifier as it also mentioned in the email mining resume by Guanting Tang et.al[33].

LSA (Latent Semantic Analysis): we already explained the high applicability of this method for textual classification operations in Section. 2.1.2. Gefan et.al[17] made a summary about several classification methods, including LSA, and pointed out using past works how LSA was successfully used in email categorization (e.g: spam/non-spam emails).

2.4 Graphical representation

Several visualizations have been deployed to assist users understanding the email data and correctly highlight the information inferred. We can divide this section in two parts: Social network visualization, and other additional graphical visualizations which depend on the type of info we want to visualize.

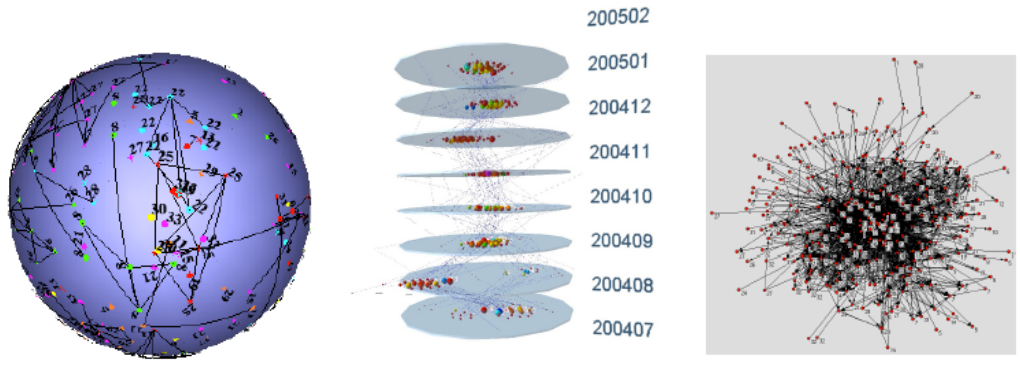
2.4.1 Social network visualization

The graph visualization is the basic and most popular structure used to visualize a social network. Formally we try to build $G(V, E)$ where V is a set of vertices (the actors) and E are the edges (the links/relations), the edges could be represented in directed or undirected form, depending on the kind of information and relations we are trying to build.

Several alternatives and modifications of the basic network graph visualization were proposed in literature. An interesting work by Xiaoyan Fu et al[15] differentiate two kind of visualization: small-world email networks to analyze social networks, and email virus attack/propagation inside a network. The first visualization is the one we are interested in, the same article mention several interesting various methods applicable for such visualization: the use of a sphere surface to reveal relationships between different contacts groups, hierarchical model displaying the centrality analysis of nodes to emphasize the nodes importance, a 2.5D visualization to analyze the evolution of email relationships over time with a time filtering option, and social circles that reflect the contacts collaborations.

The idea of the sphere visualization is to distribute evenly the nodes on all the sphere surface in order to avoid having collapsing nodes in one point of the visualization. In Figure.2.2 we show the difference between this visualization and a complex flat network graph representation, as it's mentioned in the article. The second interesting visualization is the one proposed for the navigation through different temporal ranges, a good method for time series visualization can provide a better understanding of the network behavior and evolution through time. The proposed method builds the graph evolution on different layers (plate), each layer represent the network state at a specific time, each graph layout on every plate is independent from the others, some inter layers are used to connect different layers, Figure.2.2b shows this visualization.

Since displaying a high number of nodes and edges can bring confusion and difficulties in understanding the social behaviors, other alternative graph



(a) Sphere network graph (b) Temporal layers for network graph navigation (c) Flat network graph

Figure 2.2: Network graph visualization alternatives from[15].

visualizations have been taken in consideration. In [6] a proposed approach navigates the graph from a macro visualization to a detailed micro view, in this case the macro view builds global clusters grouping correlated nodes, and a further micro navigation will display the internal elements.

Another innovative approach was presented by Sathiyanarayanan et.al [30], in this case the idea is to have a hybrid view using treemaps and Euler diagrams, although in this case the main purpose is to build a hierarchical representation of the common topics combined with the different actors that treat them, such visualization is capable to visualize both the contacts sets and their elements. This feature is very useful specially for textual analysis, Figure.2.3 shows this visualization, as we can see the circles (with different sizes) are used to represent the Euler based diagrams while rectangle of various shape represent the treemap based diagrams.

SN visualization tools

The literature offers some interesting tools and libraries to visualizing and build social networks through a network graph representation. The Table. 2.3 lists some interesting open source products with a license that permits free use in commercial settings, in the table we give a brief description of the tool

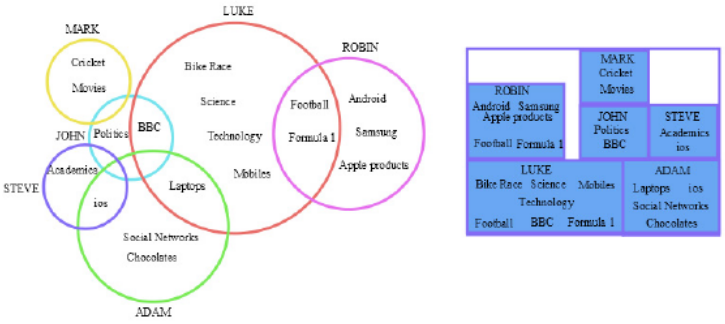


Figure 2.3: Treemap combined with Euler diagrams [30].

by mentioning the most relevant features included along with the operating system and environment needed. The developing libraries are the elements that interest us most, since we will need to use one to develop our final framework.

Cuttlefish	*Linux*
• Detailed visualizations of the network data • Interactive manipulation of the layout • Graph edition and process visualization	
Cytoscape	*Windows, Linux, MacOS*
• Perfect for molecular interaction networks and biological pathways • Integration of annotations, gene expression,profiles and other state data for the network • Advanced customization for network data display • Search/Filter options and clusters detection	
Graph-tool	*Python library*
• High level of performance (uses parallel algorithms) • Own layout algorithms and versatile, interactive drawing routines based on cairo andGTK+ • Fully documented with a lot of examples	
Gephi	*Windows, Linux, MacOS*

• Exploratory Data Analysis: intuition-oriented analysis by networks, manipulations in real time. • Social Network Analysis: easy creation of,social data connectors to map community organizations and small-world,networks. • Metrics stats eg: centrality, degree (power-law),betweenness, closeness. • High-performance: built-in rendering engine.
MeerKat *Windows, Linux, MacOS*
• Filtering, interactive editing and dynamic networks support • computes different measures of centrality and network stats • automatically detects communities and build clusters
NetworKit *Python library*
• Efficient graph algorithms, many of them parallel to utilise multicore architectures • Large ammount of data could be analyzed (multicore option make it easier)
NetworkX *Python library*
• creation, manipulation, and study of the structure, dynamics, and functions of complex networks. • network structure and analysis measures • nodes and edges can represent deferent elements e.g: text, images, XML records, time series, weights.
SocNetV *Windows, Linux, MacOS*
• Advanced measures for social network analysis such as centrality and prestige • Fast algorithms for community detection, such as triad census • Built-in web crawler,to automatically create "social networks" from links found in a given initial URL. • Fully documented online and inside the app
SUBDUE *Linux*

• Represents data using a labeled,,directed graph • Graph based supervised learning from the input data • Graph based, hierarchical clustering: discover patterns and compress the graph • Last software release 2011
Tulip *Python library*
• Dedicated to the analysis and visualization of relational data • A development pipeline which makes the framework efficient for research prototyping as well as the development of end-user applications
Vis js *JS library*
• A dynamic, browser based visualization library.The library is designed to be easy to use, to handle large amounts,of dynamic data, and to enable manipulation of and interaction with the data. The library consists of the components DataSet, Timeline, Network, Graph2d and Graph3d

Table 2.3: Open source tools for social network visualization

2.4.2 Other data visualizations

Dealing with email data is a sub issue of big data analytics and it's graphical representation [37]. Visualization is an important component when dealing with Big Data analysis, since it helps get a complete view and emphasizes the inferred information. Big Data analytics and visualization should be integrated seamlessly so that they work best in Big Data applications. The visual part should be understandable and easy to read, specially in our case, since our final purpose is building a graphical interactive web application. Conventional data visualization methods as well as the extension of some conventional methods to Big Data applications, should be taken in consideration along with the previous methods discussed for visually representing the social network structure.

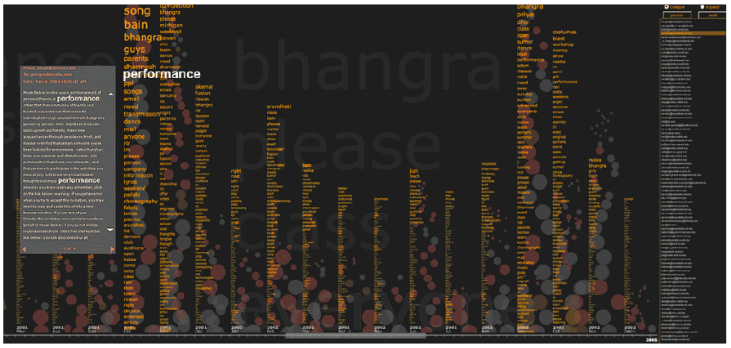


Figure 2.4: A 2D graphical representation of the text as a function of time [34].

A big part of the email mining operations we will conduct involves textual analysis. More specifically, classification/categorization of the text, is one of the options we are willing to include inside our system. To visualize this type of information we might use a treemap view [37], or as suggested from [22] a graph visualization that builds groups of related entities and applies physics behaviors like force attraction to demonstrate the level of correlation between the elements.

Apart from entities classification, another interesting form of representation we need to take in consideration is related to the study of alternatives ways of representing the data in a 2D graphic visualization. In [34] they used a novel approach for visualizing text and words inside a 2D graphic, in a way to highlight the textual content usage as a function of the time.

2.5 Email forensic tools

We analyzed some of the most important and popular tools in literature, based on 7 important macro criteria appropriate for the evaluation of email forensic tools, as already previously studied by [13], Garfinkel et al.[16], and Hadjidj et al.[19]:

Operating system: the operating system needed to run the tool

The search and filtering options: the type of searches and filters a user can apply. e.g: keywords, contact name, subject name, date, contacts importance etc.

Provided information: what kind of information can we expect to retrieve and view from our analysis of email archives. e.g: messages traffic details, contacts collaboration skeleton, context information and textual clues, general stats etc.

Email formats: the email archive formats supported by the tool as input.

Visualization method: the graphical style used to visualize the information and the data analyzed. e.g: charts, structured lists, graph networks etc.

Export format: the format of the analysis report we made and we would like to export.

Software license: the software license applied to the tool.

Search/filtering, provided information, and visualization method, are all extendable topics which might list different micro fields of study, the selection of these sub-fields is based on the information collected from the tools we analyzed, previous experiences, and common user requests.

- **Search/filtering options:**
  - Contact name: the contacts in the 'From','To', and 'CC' fields.
  - Words in context: the textual body of the message.
  - Subjects/threads name: the subject title of the messages.
  - Sending time: the time when the email was generated from the sender.
  - Filtering for a time range: filter results under a specific range of 'Sending time'.
  - Contacts relevance: a contact relevance can be represented in several ways (e.g: number of his relations).

  - Relations relevance: usually a relation assumes more importance according to the number of messages exchanged.
  - Filtering/searching combination: combine different filtering options together for a much complex searching analysis.
  - Number of subjects/threads: the number of subjects treated.
  - Concepts/topics affinity: the concepts treated by the contacts.
  - Contacts relations number: the number of possible relations of the contacts.

- **Information provided:**
  - General SN stats and metrics: in case the tool can build a SN.
  - Messages traffic information: e.g: number of messages exchanged, number of connections ... etc.
  - Contacts collaborations/clusters: build groups of contacts following a distance function.
  - Documents/attachments analysis: a deep analysis of the email attachments (e.g: doc files, images).
  - Calendar data analysis: calendar events/appointments and data.
  - Contacts and links relevance: a deeper analysis about the inter relations between the contacts.
  - Sending and receiving messages streams: differentiated analysis for the messages directions.
  - Keywords occurrences in context: summary analysis for the words usage and frequencies in the email context.
  - Geolocation: detect or mine geographic references from context or through attachment analysis.
  - Semantic analysis of the context: the study of textual meaning (Text mining related).

  - Urls/links detection in context: detect links and urls references inside the textual email body.
  - Emails detection in context: detect email addresses inside the email context.
  - Temporal occurrences detection: detect temporal references in the context.
  - Word phrases detection: extended analysis for word phrases (composed with more than 1 word).
  - Words relevancy ranking: giving a ranking score to the words according to their importance.
  - Concepts/topics auto detection: automatic detection of concepts and the treated topics inside the emails.

- **Visualization method:**
  - Network graph: usually this visualization is used to represent and a build a social network.
  - Charts and bars: to present grouped data and summary length stats.
  - Structured lists: a set of ordered heterogeneous values.
  - Geographic map: usually used to represent geolocations and messages paths
  - Clusters representations: clusters and group of elements are usually represented in a hierarchical cake visualization.
  - Dynamic (real time) interaction: a dynamic adjustment of the graphical visualization (generally for network graph case).
  - User-friendly interface: an easy, intuitive and reliable interface to use, this metric is a subjective value and its based on our personal impressions.

2.5.1 Comparison Analysis

In the Table.2.4 we present some popular tools we investigated according to the criteria mentioned before. All the listed frameworks can have the email data as possible input, although the majority prefer a global forensic analysis through the combination of different file types included in the dataset dump uploaded to the framework. We aren't going to treat separately each one of these tools, although some notable facts should be mentioned:

Intella is the most interesting tool from those analyzed regarding textual analysis and text mining, it can perform searching/filtering options on different items (emails, documents, text files) and combine the results in a cluster map view related to the keywords searches, clicking on the clusters will view the related resources and give the opportunity to retrieve the original documents, Intella allows also the search for regular expression manually defined.

Xplico is usually used for forensic analysis of traffic sniffing, this analysis can be done in a real time data acquiring. This kind of analysis can include email data and urls sniffing.

Paraben offers different forensic software packages, but it's main idea is to combine the analysis of different files and documents including emails and give a global summary investigation report.

EmailTrackerPro is basically used for tracking the emails source and destination, this will also permit to detect spam messages.

Immersion It provides the most artistic and graphically intuitive representation for the SN analysis of the email data, so speaking about a user-friendly interface the solutions adopted in this framework are the most easy and reliable ones considering the other frameworks.

Some of the features in the table have a yellow background, these features are strictly dependent on natural language techniques and textual mining

methods. In the next paragraphs we will present more details on how these features are elaborated and visualized.

A notable fact is the poor textual analysis of the email textual content (the yellow features), this comparison exposed a very limited integration of text mining techniques in the analysis of the email context. As we already mentioned text mining can be very helpful for many forensic analysis requests (e.g: community detection, spam classification).

Although almost all the software products taken in consideration make the basic searches and filtering, just few can apply a dynamic and real time application to the resulting data and rebuild the visualization (e.g: the network graph) accordingly, Immersion [21] is a good example of this. Later we will see how our framework is highly inspired from Immersion user interaction and design, since we consider Immersion a very user friendly and comprehensive software structure.

Table 2.4: Email forensic tools features comparison

	MailXaminer [20]	*Aid4Mail [26]*	*Digital Forensic Framework [3]*	*eMailTrackerPro [35]*	*Paraben EMail Examine [10]*	*Xplico [18]*	*Immersion [23]*	*Intella [25]*
1) Operating system	Windows	Windows	Windows, Linux	Windows	Windows	Web App	Web App	Windows
2) Search/filter options								
2.1) Words in context	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
2.2) Contact name	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2.4) Sending time	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2.5) Filtering in a time range	Yes	Yes	No	No	Yes	No	Yes	Yes
2.6) Subjects/threads name	Yes	Yes	Yes	Yes	Yes	No	No	Yes
2.7) Contacts relevance	Yes	No	No	No	No	No	Yes	No
2.8) Relations relevance	Yes	No	No	No	No	No	Yes	No
2.9) Concepts/topics affinity	No	No	No	No	No	No	No	Yes
2.10) Contacts relations number	No	No	No	No	No	No	No	No
2.11) Filtering/searching combination	Yes	Yes	No	No	Yes	No	Yes	Yes
3) Information provided								
3.1) Messages traffic information	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
3.2) General SN stats and metrics	No	No	Yes	No	No	No	Yes	No
3.3) Contacts and relations (SN)	Yes	No	No	No	No	No	Yes	No
3.4) Documents/Attachments analysis	Yes	Yes	Yes	No	Yes	Yes	No	Yes
3.5) Calendar data analysis	Yes	No	No	No	Yes	No	No	Yes
3.6) Contacts and relations relevance	Yes	No	No	No	No	No	Yes	No
3.7) Sending and receiving messages streams	Yes	No	No	No	No	Yes	No	No
3.8) Retrieving original documents	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
3.9) Keywords occurrences in context	No	No	No	No	No	No	No	Yes
3.10) Geolocation	No	No	No	Yes	Yes	Yes	No	Yes
3.11) Semantic analysis of the context	No	No	No	No	No	No	No	No
3.12) Urls/links detection in context	Yes	Yes	No	Yes	Yes	Yes	No	Yes
3.13) Emails detection in context	No	Yes	No	Yes	Yes	Yes	No	Yes
3.14) Temporal occurrences detection	No	No	No	No	No	No	No	No
3.15) Word phrases detection	No	No	No	No	No	No	No	No
3.16) Words relevancy ranking	No	No	No	No	No	No	No	No
3.17) Concepts/topics auto detection	No	No	No	No	No	No	No	No
4) Supported email formats	PST, OST, EDB, MBOX, etc	Remote server, EML, MSG, PST, MBOX, etc.	PST, OST, Raw, EWF, AFF	AOL, AOL, Web Mail Gmail	PST, OST, Thunderbird, AOL, etc.	Hotmail, Gmail	MBOX, Gmail, MS Exchange	Hotmail, Gmail, IMAP, MS Exchange
5) Visualization method								
5.1) Network graph	Yes	No	No	No	No	No	Yes	No
5.2) Charts and bars	Yes	Yes	Yes	No	Yes	Yes	Yes	No
5.3) Structured lists	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
5.4) Geographic map	Yes	No	No	Yes	No	No	No	Yes
5.5) Cluster map	No	No	No	No	No	No	No	Yes
5.6) Dynamic interaction	No	No	No	No	No	No	Yes	Yes
5.7) User-friendly interface	High	Medium	Low	Low	High	Medium	High	High
6) Export format	HTML, pdf, csv	PDF, HTML, PST, MBOX, CSV, XML	EML, PST, TIFF, PDF, MSG, HTML.	Excel, HTML	PST, MSG, EML, HTML/XML report	N.N	None	HTML, PDF, CSV
7) Software licence	Commercial	Commercial	Open source	Commercial	Commercial	Open source	N.N	Commercial

Chapter 3

Proposed approach

Theoretically our framework is based on the integration of different concepts from study fields of interest. In the previous chapter we made a research study on some important matters, and pointed out interesting features and topics already taken in consideration by past works along with new innovative techniques, specially related to text mining processing, that we believe can significantly improve a forensic investigation.

The original data format is given in an unstructured representation, converting this original representation of the data in a comprehensive, usable, interactive, and navigable data analysis interface, involves a gradual processing study and implementation of different phases.

In this chapter we will treat separately conceptually the processing phases needed to build the final framework. Starting from the preprocessing phase and the preliminary yet essential procedures needed for the textual mining models we adopted in email context analysis. Further we will explain the methodologies used to extrapolate information and mine the email data. These phases will guide us through the final definition of our system. On the final section of this chapter we will list some possible future implementations and improvements to the final product. These phases are summarized in Figure 3.1.

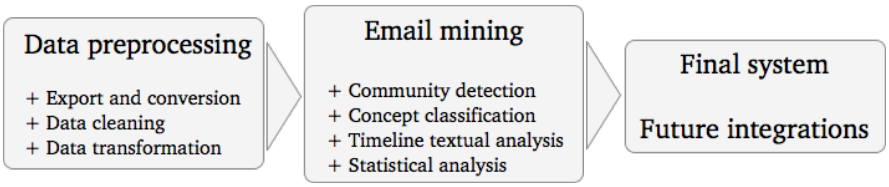


Figure 3.1: Framework formulation phases

3.1 Data preprocessing

Email data backup formats are generally incomplete and contain a lot of noisy textual parts, which may be irrelevant and cause negative results when processed together with the meaningful data. This phase is a very important preliminary step before proceeding in the actual data mining process. The main objectives of this phase is transforming the raw data into an understandable format representation. Later when we will discuss the actual model the importance of this phase will become more clear, and let us better understand the importance of these preprocessing elaborations, specially when treating the textual content of the email archive. Some of these operations are essential, while others will evidently help us maximize the efficiency of the algorithms that we will use further.

This chapter will outline the methods used in data preprocessing in three sub fields: data conversion, data cleaning, data transformation and data reduction.

3.1.1 Data export and conversion

Email archives can be exported in various formats, this will create a unique file built on the mailbox selected. This operation can be done using some basic custom email clients, for example Outlook uses a tool called the Import/Export wizard, and google lets you download all your personal data (contacts, calendar, fotos etc) including your email box archive through the takeout portal.

Different clients use different exporting formats, in case of commercial

```
From MAILER-DAEMON Fri Jul  8 12:08:34 2011
From: Author <author@example.com>
To: Recipient <recipient@example.com>
Subject: Sample message 1

This is the body.
>From (should be escaped).
There are 3 lines.

From MAILER-DAEMON Fri Jul  8 12:08:34 2011
From: Author <author@example.com>
To: Recipient <recipient@example.com>
Subject: Sample message 2

This is the second body.
```

Figure 3.2: Example of a typical contents and structure of a mbox file

softwares, such email archive formats can seriously limit our further analysis. A very common and generic file format used to hold email messages collections is MBOX. The final .mbox file will contain a list of textual concatenated messages, usually each message starts with a From word followed by the header metadata (as already discussed in Section. 2.3), this kind of file storage generation could be directly accessible by individual users, unlike other commercial formats. In Figure. 3.2 we have a typical example of mbox file.

Although there is the possibility to operate on a vast set of email formats and try convert them to a common representation, for a further use as input to our framework, we choose to use mbox as the only acceptable data dump file format. This is due to the fact that conversions may lead to possible data loss or incorrect reconstruction. Future implementations can reconsider this problem and try find a suitable method to extend the supported file formats, we will elaborate this aspect in the dedicated section for the future integrations Section. 3.3.

3.1.2 Data cleaning

Meanly in data cleaning we have to deal with two different sub fields: the missing data, and the unnecessary information. Missing data is a very common phenomena in datasets, the basic approach to deal with such problem is by first understanding the reason why we have such missing information and if their is a specific pattern that is common to the parts where this happens. A possible solution is the complete remove of the missing data from the correct files also, this technique is particularly applicable if we have little effects on the final results. Otherwise if we find or discover a particular pattern, a possible solution is filling the missing values by following correct data examples, for instance with common values or average values.

A very interesting field which is missing in many occasions in email archives is the Reply-To field, this header attribute is used to indicate where the sender wants replies. Unfortunately, this value is very ambiguous, since we have many possible addresses representation (e.g: group names). In addition to this difficulty, replies have different form of representation in the textual body. We decided to treat the dataset as its always missing these type of headers, and we will try to rebuild the replies path through textual filtering techniques.

An important question that arises while observing the emails data is How we can decide whether two different emails A and B refer to the same email subject?. First we need to understand how users usually reply to emails. Although such behavior could be exhibited in many different ways and patterns, some custom operations are very common:

- The original subject name is modified and will usually contain a 'RE:' or 'Re' before it.
- The previous email message body will have '¿' before each line in it, or a prefix word like: ' From:', ' wrote:', or 'message' will denote the begging of the previous message body block.

Giving these common patterns, we will actuate a data cleaning process to

filter and remove the extra textual parts in the subject and classify all related emails under a common subject title. Additionally, we need to remove the textual body of the emails that have been replied to (the second point of the previous listed items), this part of data cleaning is essential to avoid textual redundancy of same email context, further we will show how these factors may have negative effects when applying text mining operations.

In addition to this, we need to pay attention to the textual content of emails, since they might contain a lot of irrelevant parts, we should filter and take off, so we make sure we will not get negative future repercussions. The information relevancy is strictly related to the kind of analysis we are planning to do. Some common operations used for this case are:

Remove redundant information: moreover still dealing with the problem of duplicates, a very common example for this case, are the final or entry segments that contain the address and the contact information about the sender. This phenomena is vastly common in many emails. When analyzing the textual content of a large archive, the high occurrences of such words could suggest a big relevancy to these type of information, and give a high relevancy (score) to them at the expense of other interesting information, so it's highly suggested to remove and ignore these data from the final analysis. Other redundant segments could be reveled after some data-verification tests, and would suggest the introduction of additional ad-hoc rules to exclude them from the data reformation output. This final aspect will become more clear further, when we are going to show an effective example of data uploaded to the system and the data cleaning elaboration made in that case.

Context normalization: to normalize a content we might for example need to deal and remove non-alphanumeric characters or diacritical marks. In some occasions a correct normalization requires some advanced knowledge on the input data, for example the language used in the original source. In case of email data, it is very likely to have to deal with unwanted HTML tags (e.g: if a table is generated and

included in the email body). In addition it's also important to know the language used, this will help us define the set of stop-words to cut out from our future mining operations and textual elaboration. Beyond these ad-hoc data cleaning procedures other more general operations are taken, like removing special characters (e.g: break lines tags) and excessive white spaces between the words.

3.1.3 Data transformation

A fundamental preprocessing part is transforming data into a suitable form comprehensive to the final application and ready for further analysis. Users that reply to a specific email message, portrait the same situation we have when a group of people discuss a specific common thread. One of our objects is to reconstruct the timeline and the actors of such discussion, this can be done by observing the From, To, CC, Date, Subject, and Body of the email messages. The From, To, CC fields help us understand the actors involved and the relation direction, formally: if $f \in FROM$, $a \in TO, CC$ the relations generated are all the possible combinations $f \rightarrow a$. The Date field helps us order and schedule emails in a specific timeline. In order to decide the associated subject of each email we will use the previous data cleaning techniques.

Our final object is to transform the email archives into two basic data set representations: nodes and edges. The nodes data set will list all the actors (email contacts) present in the uploaded archive, while an edge will basically contain the following attributes:

- Origin: the sender of the message
- Destination: the message receiver
- Time: the time when the message was generated
- Subject: the message subject (thread title), after the cleaning data operations

- Content: the message content, after applying the cleaning data operations

In our work we made a distinction between two basic definitions: email and message. The email is the traditional textual format we commonly receive and can view through traditional clients, while a message is another representation we use to denote a 1:1 relation between the sender and the receiver. This example might help understand this relation: If we have an email E and it's fields are: [From: F, To: T, CC: C, Date:D, Subject: S, Content: C] with $F = [f_1]$, $T = [t_1]$, and $C = [c_1, c_2]$, then we will have 3 different messages, all with same subject S, date D and content C, but with a different origin and destination: m_1 = [Origin: f_1, Target: t_1], m_2 = [Origin: f_1, Target: c_1], and m_3 = [Origin: f_1, Target: c_2].

Since the final data form needed by the model (nodes and edges) are independent from the original data representation, the data transformation could be applied to different situations such like emails. All we need to do is adapt the edges and nodes (with their attributes) to the type of data treated. Some interesting future applications propose the integration of social networks data, all we need to do is redefine the past basic concepts (e.g: edges, subjects, sender, receiver ...etc). These future possible applications will be discussed later in the next sections.

3.2 Email mining

Is the application of mining and mathematical methods to explore the uploaded email archives, this will let us understand new knowledge, new patterns, and predictions about unseen data. We will divide this section in 3 different fields: community mining, topic classification and timeline textual analysis.

3.2.1 Community detection

As we already mentioned in the previous Section. 3.1.3, our dataset will be composed from basically two different tables: nodes and edges. These two tables are all we need to build our network graph representation for contacts collaborations and relations. We define and generate two different types of social networks:

Collaborations/relationships network: this representation is used to visualize the collaboration patterns of the contacts based on the shared email subjects. So we are interested in building and showing relations between the contacts independently from the owner of the email archive, that's why we will remove the owner contact from the list of network nodes. In order to build this network typology, we need to ignore all messages under the same subject title which have been sent just on a one and only occasion (one date), this means that these type of subjects have no replies, and thus do not represent a discussion thread. This method implicitly removes spam emails from the visualization (although it's out of our interest to further analyze them separately), that's because almost all spam messages come with different titles and with a one shot type of messaging and have no replies from the final user.

Messages traffic network: in this case we make a distinction between senders and receivers, this will emphasize the analysis on the amount of messages and the related traffic direction that each node do. In this case the network graph will also contain the contact of the email archive owner, this will help users monitor the different messages received and sent by all the contacts, along with the owner of the email archive messaging activities. In this network typology spam emails (or emails with same behavior) will not be excluded from the final representation, this fact can help us for instance detecting high email traffic from specific unrecognized contacts.

Each element of the nodes table, previously generated in the data transformation, will represent a different actor for our final social network representation, the owner contact will not appear in the community network representation. On the other hand, the nodes table will stay intact with no modifications for the 'Message traffic network' generation.

Since the edges table will basically contain: origin, target, time, subject, and content. The edges of the network will include all the unique combinations of origin and target values, the table might contain same edges (origin,target) which relates to a different subject and/or time. We define the weight of an edge as the sum of all these possible rows in the table. In case the social network we are trying to build is the 'collaboration network', the edges weight value will not take in consideration those where the owner contact address appear. Giving this definition of network edge, a degree of a specific node is the sum of all the edges connected to it, and for the 'Message traffic network' case we will have the ability to separate the degree value in: in-degree and out-degree respectively for received and sent messages. In addition we will define the 'value' of a node as the sum of all connected edges weights.

3.2.2 Concept classification

While it's possible to use traditional keyword searching techniques for example to detect if criminality related words are mentioned (e.g: drugs), these techniques are inefficient and may not give any significant results or suspected anomalies in the data analyzed, since suspected users usually do not explicitly use such words, instead other expressions and encrypted messages are preferred, which might hide different suspected meanings.

Classification techniques are more robust to noise and dimensionality, in addition the final results are more precise, and can easily elaborate large amount of data, otherwise much more difficult to analyze with manual ad-hoc searches.

For email messages, a text based classification algorithm helps us clas-

sify emails in different categories, some might turn out to be anomalies and unconventional categories, if compared to the type and expected usage of a particular email address, for instance using the work email for personal private use and duties.

Natural language techniques (NLP) help us understand, elaborate, and mine the textual content of the email. A very common and successful approach for textual classification is LSA (Latent Semantic Analysis), we already talked about this powerful tool in the textual mining techniques background Section. 2.1, we will use it along with TFIDF as text weighting algorithm. We can summarize the LSA process in these steps:

1. **Building the corpus/collection of documents to use as input:** the corpus that we must generate is a list of all the different emails in the archive, we should pay attention to text redundancy, and avoid it. This is done in the preliminary phase of data cleaning as we already mentioned in Section. 3.1.2, the main reason for the existence of this problem is due to the presence of reply emails, usually these messages copy the text of messages that they are replying too, along with the actual reply message. We will filter and exclude these contents using the procedure we already mentioned in Section. 3.1.2, and next populate the corpus with these filtered (cleaned) documents (emails).

2. **Building the word phrases dictionary and removing stop words:** we will use the n-grams model to build the set of word phrases, and we choose a granularity of $n = 3$, which means that the maximum word phrases length we might have is 3 (e.g: new york city). It's very common to use trigrams models especially when the available training data is limited, and this particular n value proved to be very successful in detecting important and relevant word phrases, and in addition the data elaboration time and complexity is less expensive, 4-gram and 5-gram models are used when the available data is very large. We should exclude stop words from the dictionary words, stop words are extremely common words which have small semantic relevance to the final anal-

ysis. This set of words is strictly dependent on the text language, and can be updated with additional ad-hoc words. The proposed framework will leave this as an open option and will let the user choose the language and manually add other irrelevant terms.

3. **Applying a TF-IDF text weighting algorithm:** Its the combination of term frequency and inverse document frequency metrics. This value will be high when a term occurs many times within a small number of documents, while we will get a lower value when a term occur fewer times in a single document or many times in many documents. For a further and mathematical definition of TF-IDF, we send you back to Section. 2.1. This step will create a matrix (terms x documents), and each cell will contain the tf-idf value.

4. **Applying a matrix decomposition scheme SVD (Singular value decomposition):** giving the matrix of step(3) we will construct a low-rank approximation of it using SVD. This algorithm will decompose the matrix to three different matrices. SVD will decompose the original matrix to a lower rank K, this value is generally chosen to be in the low hundreds when having a very high rank. For our case, this value is chosen ad-hoc according to the data analyzed, by certifying manually the results validity. We choose this approach mainly to optimize the elaboration time. Further in section 3.3 we will give a possible more sophisticated approach to deal with this problem and automatize the value of k.

The final step of LSA (decomposition) helps us represent and classify the original documents into a new set of documents, these new documents represent different concepts. For each concept (cluster) we will retrieve the set of terms with higher scores, these terms are the most representative words of that concept, and might give an idea about the possible common subject that pool these terms.

Now that we have different clusters of concepts, we need to know the

clusters affiliation to the network elements. First we need to redefine the set of documents. Two different approaches could be adopted:

Nodes: each node will have a different document that includes the context of all the messages that he treats

Edges: each edge (relation between two contacts) will be represented as a different document containing all the messages exchanged between the two nodes.

Both techniques will consult the words x documents matrix, and get the collection of vector space representation according to the documents needed. This will let us apply cosine similarity operations between clusters space vectors and Nodes/Edges representations. We decided to integrate both approaches in the framework, and let users decide the better representation according to his needs.

3.2.3 Timeline textual analysis

The aim is to visualize email content over the time, so we want to associate a list of the most used word-phrases for different time values, and rank these words according to their importance. This can help us answer the question what are the words i use most on different time periods?.

The time period span value is correlated to the maximum possible range of time $d = time_{last-email} - time_{first-email}$. Although as also suggested by [34], the most common and beneficial way is to represent time periods in months and years.

To weight and rank the word-phrases we will again use the TFIDF algorithm, although this time the set of documents will be categorized according to periods of time, we can define the processing steps like:

- **Building the corpus/collection of documents to use as input:** giving the earlier email sent time T_0 we will convert this value to a $F_0 = (T_0(year), T_0(month))$ representation and build a sequence of F

values, such that $F_i = F_{i-1} + i$-months the last value of the series will contain the year and month of the email with the higher time. For each period of time we will associate the text of all the related emails. As we did with the concept classification, we will have to again apply the preliminary phase of data cleaning with the same previous procedures as we already discussed in Section. 3.1.2.

- **Building the word phrases dictionary and removing stop words:** the same options and pre-configurations used in the concept classification will be applied for this case too.

- **Applying TFIDF algorithm**

At the end of the last step we will have a sparse matrix words x documents, we will convert the final matrix to a dense version (in order to remove all the zero values), since usually the order of the matrix is very high. After we have a new dense representation of the matrix we will sort it according to the tfidf values. As a result, each different document (time period) will be associated to it's set of ordered terms according to their relevancy.

The final matrix will contain different word-phrases with different gram dimensions, a very common scenario which might happen, is having different word-phrases with different gram value but with the same tfidf score, For instance: 'new york' and 'new york city'. The fact that these words have the same score, suggests that they actually appear almost in the same places: 'new york city' and 'new york' are never mentioned separately in different context parts. This situation will cause the presence of both word-phrases when we generate the ordered list for each time series value. To avoid this, we decided to keep only the higher gram value word-phrase and exclude all the others from the ordered list. Before excluding the lower gram word-phrases we will make sure they have the same score of the bigger gram word-phrase which includes them, otherwise we will keep them in the final ordered list.

3.2.4 Statistical analysis

A statistical analysis of the network generated from the email archives analyzed can expose a great deal of information. The possible stats and information we can deduce could be divided in two major sub fields: Graph network metrics, and contacts and links attributes.

Graph network metrics

In this case we talk about general statistics about the graph network generated from the 'community detection' phase, and we introduce some of the most common graph metrics along with new associated information:

Node degree: the number of edges incident to a specific node. In case the graph network is a 'Directed' one, we can distinguish between in-Degree and out-Degree for respectively entering and output edges.

Edge Weight: A numerical value assigned to every edge to denote the number of messages exchanged between two nodes.

Node strength (value): The sum of weights attached to a specific node.

Summary stats: number of all the nodes and edges, and average value calculation about the previous degree, weight and node strength.

As we already mentioned we define the typology of graphs we are building as temporal graphs. Since the time dimension is a fundamental search parameter, it will introduce dynamic modifications to the graph structure and all the above metrics must also be dynamically calculated as soon as the graph changes his structure.

Contacts and links attributes

In addition to their representation in the graph network model, contacts and links contain other important information which cannot be deduced only by their graph network structure:

Messages sent and received: the number of messages a specific contact send and receive, or the number of messages exchanged between a couple of nodes (link).

Messages as a function of time: create a distributional function for the messages (sent and received) as a function of a specific time range (which might also dynamically change). The time axis should be represented in a set of discrete time series values, the granularity and set of values to consider on the time axis depends on the difference between the higher and lower value of the time value parameter.

Contact domain: the domain of a specific contact can be easily deduced from his address, it's the second part of the address, right after the '@' character. Assigning the domain of all the contacts, could be also considered as a 'Contact categorization' operation, since it will divide the contacts in to several groups that share same domain name.

Subjects: the set of subjects where the contact or link got involved, the previous community detection elaboration already classified the contacts according to the common threads they treat. Each different thread contains a sub set of messages under it. A common way to organize those is by ordering them according to the sending time.

3.3 Future integrations and optimizations

Almost all the components included in the system and treated in previous sections could be optimized and extended with new features. In this section we will propose again same main topics previously analyzed, and give hints on what are the possible additional improvements that could be applied, and what are the expected results.

3.3.1 Data preprocessing

Supported email formats: The actual system supports only MBOX files type, the set of supported formats could be further extended to very popular proprietary formats such like PST, a possible solution is to create a converter tool which takes various possible email formats as input and generates a common data-row type of file storing all the data of the archive as output.

Data cleaning: Redundant and excessive information in emails can vary according to the type of data in the archive uploaded and analyzed, for example for some personal email archives we can find at the end of each email a sentence like 'Sent from my iPhone'. A good solution to this is to give users the option to add adhoc regular expressions rules or textual phrases to remove. Another valid solution, is to apply a pre text mining procedures to point out highly frequent words and phrases and let users decide what parts should be excluded from a further textual analysis. A pre-analysis is highly dependent on the quality of validation operations we made, usually with a large quantity of data we can get a more accurate textual patterns that occur and need to be removed.

As we mentioned before, replying emails are a very important subcategory of text which needs an attentive filtering procedure: since a lot of these emails contain also the content of the emails they are replying to. To remove these parts, additional techniques could be adopted and new patterns could be investigated, in order to guarantee a larger coverage of different situations. New techniques must guarantee no information loss, and non excessive elaboration time and resources.

3.3.2 Email mining

Attachments: email attachments are very frequent and might contain documents or images which can hide important information. A future integration of this analysis might involve images, videos and general

documents analysis for suspicious materials. These operations are normally very expensive, specially if we want real time results and dynamic filtering application on the data. However, studies on the forensic analysis of such files have confirmed the importance of this analysis. This analysis should be also applied to the related meta data of the attachment, along with the actual inner content. For instance: studying the meta data of attached images might reveal the place and time where the picture have been taken.

Geographic location: geographic instances could be detected by scanning the textual content for geographic occurrences, this involves additional text mining operations and textual semantic deductions, which might also use a geographic ontology, as already suggested by some studies [24] [36].

Temporal instances: although the proposed framework guarantees a timeline visualization of emails, this one is based on the actual sending time in the emails header. An interesting integration is to extend this search to a new timeline based on the temporal instances in the email context. This kind of analysis will let us also monitor the temporal references in the email body, which might hide significant dates frequently occurring.

Calendar data: the calendar data can include all type of previous and future planning events. The analysis of this data is very useful, we can for example track the past appointments made by a specific contact and later using this information to track the positions and places where a user moved to. We might combine these new information with other evidences extracted and previously mined, to form a more complex and sophisticated forensic analysis.

3.3.3 Textual mining techniques

Vocabulary terms: To build this dictionary we use n-gram models and exclude the set of stop words from the final terms dictionary. Both

these operations could be elaborated differently: our approach expects a value $n = 3$ for the n-gram model, the optimal selection of this value is related to the type and quantity of textual content of the email archive. Therefore a more deep analysis of this aspect should be taken in consideration. Testing the system with different gram values can highlights different performance results, and help us decide the better solution. Building the set of stop words is another crucial part in the definition of the final terms dictionary, the proposed framework already creates it according to the textual language and a manual addition of others terms. Although, a valuable solution could be to involve automatic detection techniques using tfidf, since this method can detect and give a limited significance to those terms that appear frequently and in many different documents.

Concepts classification: Finding the most suitable number of concepts is a fundamental aspect which needs a deep analysis and study. The number of concepts to establish will represent the k rank of the decomposition operation (SVD) of the LSA algorithm. A possible solution to this problem is to test LSA with different values of k and to check how much the concepts inferred are unlike each other. To check the diversity of the concepts, we can compare the most representative terms of each concept and make sure that they have few terms in common (the union between the two sets). We will select the k with the higher score based on this diversity metric, and use it as a low-rank value for SVD phase.

Word-phrases frequency over time: we have already mentioned before the n-gram problem in Section. 3.2.3, which will let us exclude lower gram word phrases and keep the higher ones. However we didn't actually covered all possible inconvenient situations. Another problematic situation is having two different word-phrases with same score but neither one of them includes the other, and they might have only one

word in common. A possible cause for this, is because text weighting schemes clean all the irrelevant words which as a result compact the final textual content in a more dense form with new vocabulary unexpected terms. A further investigation of this phenomena is highly suggested in order to achieve a higher final performance results.

3.3.4 Social network analysis

The current proposed approach is built around the analysis of a collection of email messages, although a future integration of other type of inputs is possible. The framework skeleton is adaptable and very feasible to such integration.

Generally, a social network is defined as a network of interactions or relationships, where the nodes consist of actors, and the edges consist of the relationships or interactions between these actors. Although in this case we are observing the naturally basic definition of the network. Our approach wants to take advantage of other SN aspects and focus on the data treated by the actors and the way they interact with each other.

All we have to do is redefine the basic data structure components according to the type of data we want to analyze, basically: the nodes and edges for the graph network, the threads/subjects and the relation characteristics between the different nodes. Once we redefine all the basic concepts all the other components will work accordingly, which will insure us the same forensic analysis elaboration like the email case.

These basic elements could be constructed in several ways and based on different SN behaviors, in fact social networks outlets provide a number of unique ways for users to interact with one another such as posting blogs, or tagging other contacts inside images. these kind of interactions are considered indirect, and they can provide rich content-based knowledge which can be exploited for mining purposes. In fact, any web-site/application which provides a social experience in the form of user-interactions can be considered to be a form of social network, thus a possible input for a further adaptation

to the framework.

For instance, a possible scenario could involve the analysis of private Facebook accounts data and messages, in this case we may define a post on the profile wall as a possible subject, and all the related comments as replies from FB users that cooperate on a same thread discussion. The same situation could be also applied to the Twitter case, although it's important to note that in this situation the set of possible users that can reply (retweet) is the whole twitter network, which makes the problem much complex.

Chapter 4

Our framework

The final framework released is an email analysis software which focuses on analyzing forensic aspects and inferring social behaviors by taking as input a collection of email messages in one large archive.

This framework is particularly useful to report and investigate email data archives and deduce new information that might help users and inspectors understand the messages and their content. It might revel interesting clues and anomalies, as well as for example detecting email security violations.

The framework was developed in a web application format, and it could be consulted through the major important web browsers. We provide a demo version at the address: `http://smartdata.cs.unibo.it/emailAnalytics/`, which is hosted by a local server in the university.

This chapter will discuss the primary parts of our framework focusing on the implementation techniques and the architectural aspects, along with the graphical and visual interaction parts.

4.1 Architecture

To correctly implement all the framework parts, we need to define a comprehensive scheme and a well defined elaboration road map. The traditional and most commonly used application architectures (specially for web ap-

plications) are the traditional layered architectures and the 'Model View Controller' schemes (MVC). These kind of architectures might have some problems and things does not turn out always as planned. Some of the most common problems are:

- A very tight coupling between UI (User Interface) and business logic and between business logic and database logic, specially when using the traditional layered architectures.
- Systems build on such skeleton could be hard to maintain.
- These kind of architectures are less flexible for new additions and future updates.

Taking in consideration these negative aspects, we will introduce a new alternative interesting and useful architecture map used specially for framework design which can overcome these kind of problems: the 'Onion architecture'. A definition of the main aspects and logic parts of this architecture will be explained in the next section. Right after that we will apply such model to our framework and discuss the main adaptions, taking in consideration the theoretical system background of the previous chapter (see Chapter. 3).

4.1.1 The onion model

This type of architecture is mostly suitable toward an object-oriented programming, and it puts objects before all others, which makes it very successful specially for large projects, were usually data objects are frequently used. The skeleton form (as it's also suggested from the name) is built like a set of listed layers in a circle shape, as we can see from Figure 4.1.

The first three inner layers of Figure 4.1 are defined as the core layers. Let's list all the layers and give a description for each layer:

1. **Domain objects:** all the domain objects will be presented at the very core of the architecture. We will restrict the definition by keeping just

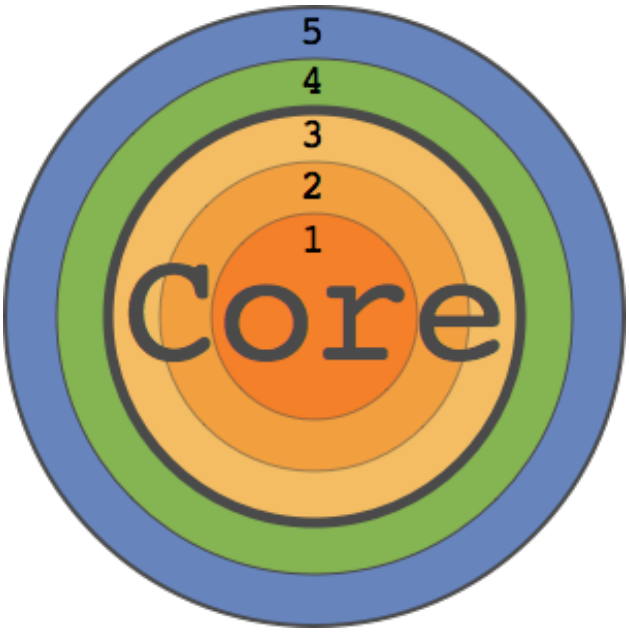


Figure 4.1: The onion architecture: (1) Domain objects, (2) Domain services, (3) Application services, (4) Extra app services, (5) Interface & infrastructure

the properties of the objects and not any extra piece of code which interacts with the database or any other layer.

2. **Domain services:** the most common operations like: adding, saving, or deleting. All these basic operations should go in here within interfaces. it's important to notice that the service interfaces are kept separate from their implementations, which shows the loose coupling and separation of concerns.

3. **Application services:** all the implementation of Interfaces defined in the previous layer service Interface layers comes here. This layer acts as a middleware bridge to provide data from the infrastructure to the final user interface.

4. **Extra application services:** additional applications service which could be considered secondary and with a lower priority.

5. **Interface & infrastructure:** this is the outermost layer of onion

architecture, it deals with the Infrastructure needs, and provides the implementation for the repositories interfaces. Only the infrastructure layer knows about the database and data access technology, while other layers will ignore all about from where the data comes and how it is being stored.

The fundamental and basic rules of this architecture are:

- Implementations and code written on a specific layer can depend on layers more central, but it cannot depend on higher layers.

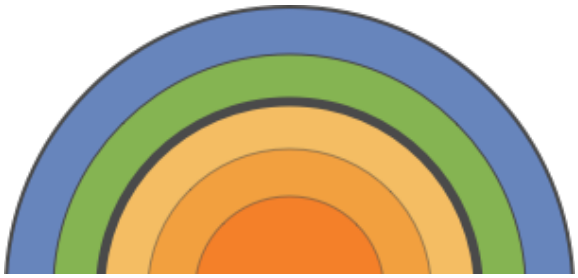
- The Inner layers will define the interfaces, while the outer layers explain the interfaces implementation procedures. This means that all the core code of the application can be compiled and run separately from the rest of the infrastructure. This fact optimizes future updates for the system, specially when we treat big applications and business frameworks.

- The fact that databases are externalized (located on the outer layers), makes the whole system independent from the kind of files and DB we are elaborating in the application.

4.1.2 Our framework architecture

Following the definition of the onion architecture of the previous section, we define the layers of our framework and the elements contained in them like in Figure. 4.2. Let's outline and describe the most relevant parts of each layer:

Domain objects and behaviors: The main basis is a graph network, which might be in a directed or undirected form. So the primary objects of the system are the nodes and the edges. The fundamental attributes of these elements are the size and the color, and it should be possible to update and modify these values dynamically.



Core Level (1): Domain objects and behavior	***Level (4) Additional application services***
Graph network visualization	Semantic analysis of the text
Nodes and Edges (users and links)	Extract urls and emails from context
Size and colors for nodes and edges	Sankey Diagram visualization
Node groups and clusters view	Filter according to keywords
Dynamic reconstruction of the network graph	Filter the graph according to topics (subjects)
Directed and undirected graph option	Upload multiple accounts data
	Geographic occurrences detection in the context
Core Level (2): Domain services	Temporal retrieval from the context
Enabling the selection of subgraphs (nodes and edges)	Calendar data analysis
Repository Interface for saving and visualization export	Files and documents analysis
Filter and search options	Images metadata analysis (e.g: for geo detection)
Filtering/searching combination	Integration with social networks
View information about the entities	Setting the graph network properties for SN case
Show statistical information in chart and bars	
Multiple graph views for different typologies	***Level (5): User Interface, Infrastructure***
	Web-app interface with high compatibility in multiple browsers
Core Level (3): Application services	A multi panel structure(Visualization, filter and Informations)
Upload an email list archive	Animated graph network for nodes and edges drag possibility.
Filter according to edges weight	Slider type input for time range definition.
Filter according to nodes degree	Integration in mobile devices
A time range filter for the data (Timeline matching)	SN data backup archives upload
Display general information about the graph properties	.MBOX format for email archives
Display detailed info about graph selections	.PST, .OST formats compatibility.
View data traffic details	.PDF, .JPG export formats of the view
View ordered/ranked lists for links and nodes infos	View demo examples for use cases
Textual mining and context analysis	Automatic data upload from account submitting
Scan for general words occurrences	Export statistics and info in structured list formats (e.g: .CSV)

Figure 4.2: (A) The onion architecture structure legend (B) The framework layers and contents

Domain services: One of the most important services that should be guaranteed is the possibility to filter the domain objects of the graph network according to some values given to their attributes, different filters (on different attributes) need to be combined in case the user request it. Secondly, we should ensure a dedicated information section for the entities that form the basis of the application, the visualization techniques could involve the use of graphs and charts.

Application services: In this layer we mention the application surface of

the domain services. The basic filtering operations to apply are those on the fundamental attributes of the entities, the degree and strength for the nodes and the weight for the edges case, in addition the temporal filter, which will be globally applied to all the system, is a crucial option to take in consideration. A big part of the generated information relies on the textual mining techniques and the context analysis operations defined.

Additional application services: All the previous layers formulate the core and essential parts of the final system. In this layer other additional services are mentioned, regardless the importance of their integration, the system can still operate and perform his basic activities. Theoretically we discussed a big part of these listed items in Section. 3.3, we will specially focus on new services that take advantage of textual mining procedures, for instance to extrapolate time and geographic occurrences.

User interface and infrastructure: this layer mention the actual data formats for the system input and for exportation and reports summary. In addition here we will define the UI aspect and build the graphical visualization of the application.

4.1.3 Modules and infrastructure

The framework is basically composed from the integration of several submodules and programming languages. The two parts involved are the server and the client. The primary initial operations and the most complex operations will be handled by the server. Allowing the server to handle the hardest operations, will permit a faster data elaboration for the user on run time execution on the client side.

All the server operations will be conducted from two python modules: dataGen.py and nlpProc.py. The data elaboration, which mainly includes the preprocessing operations (data export, data cleaning and data transfor-

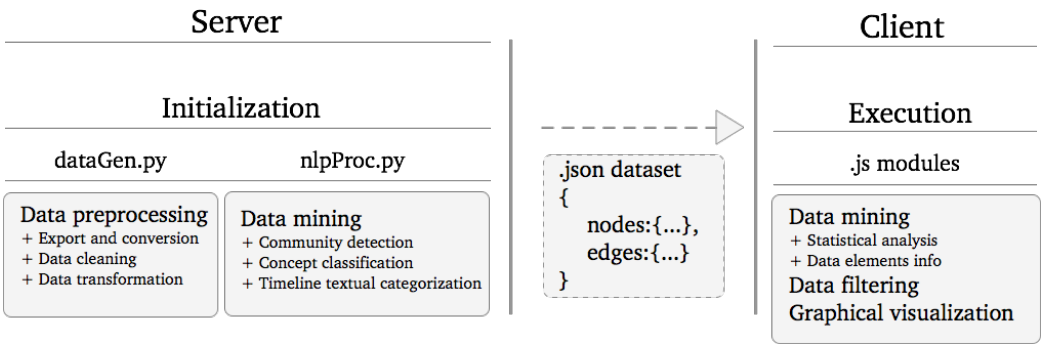


Figure 4.3: Framework infrastructure scheme, modules, and operations handled

mation) will be all managed from the dataGen.py module. On the other hand, the textual operations and the natural language processing procedures will be managed from the nlpProc.py module.

The client side is constructed from a collection of Javascript, HTML, and CSS files, used for building the graphical interface along with the model and behavior of the application. Some of the data mining operations will also be made at this point on the client side, this includes some statistical analysis and elaborations on the network graph elements.

The communication between the two parts will take place through POST requests and the type of data exchanged will be in .json format. Although, the client will not directly communicate with the .py modules of the server, instead the web application and JavaScript modules call the operations through POST requests to a .php file, who will redirected the request to the correct python module. The scheme in Figure. 4.3 presents this infrastructure and modules cooperation as just described.

In the next sections we will analyze separately the initialization and the run-time processing phases, and the related sub activities made on those phases. Next we will talk about the graphical aspects and introduce the main visual elements used to represent the information.

4.2 Initialization

The initialization phase of the framework need to elaborate and execute the first massive mining operations on the data passed as input, and then to transmit the results obtained. This will happen right after it's transformation in a suitable form for a further elaboration by the client side. As we can see from Figure. 4.3, we have preprocessing procedures along with data mining operations to execute at this phase from both the python modules dataGen.py and nlpProc.py. In this section we will show the implementations and the applications made for both data preprocessing and email mining.

4.2.1 Data preprocessing

The preprocessing data elaborations: data export/conversation, data cleaning and data transformation (see Section. 3.1). Must be executed at the very beginning, the final result produced from this elaborations is the dataset that will be used as the original and basis data representation on our application. As soon as the data elaboration comes to an end, the server will send back the dataset in a .json format to the client, who will consequently store and use it for the run-time elaborations.

Since the final dataset generated must contain both the network categories (directed and undirected), we need to apply the community detection procedures at this stage in order to build the appropriate dataset for the undirected graph network (Relationships network). More precisely the community detection phase will be integrated as a part of the data transformation stage in the preprocessing phase.

4.2.2 Email mining

The email mining operations that should be handled at the initialization phase are: community detection, concept classification and timeline textual categorization (see Section 3.2). All these elaborations need a high usage of resources and computational time, so it's highly suggested to let the server

handle this operations at the system initialization phase, and operate minor mining procedures on the application run-time.

Community detection

Since the final dataset generated as output from the initialization phase must contain both the network categories (directed and undirected), we need to apply the community detection procedures at this stage, in order to build the appropriate dataset for the undirected graph network (Relationships network). The community detection phase will be integrated as a step to execute in the data transformation stage in the preprocessing phase. In order to achieve this we define a method called collaborativeSubj(), which will build a dictionary list for all the subjects considered as collaborative threads. The graph elements: nodes, and edges that contain these contacts, which will compose the undirected graph dataset, need to be involved in subjects that appear in the collaborative subjects dictionary.

Concept classification

At this phase we will build the concept clusters and the list of terms (word phrases) more representative for each different concept (with the higher relevancy). In addition to this, we also need to associate each element in the graph network (As we theoretically explained in 3.2.2) to it's corresponding concept. These operations need textual mining analysis, therefore it will be a duty of the nlpProc.py module. The final result returned to the user, will be integrated inside a .json format.

Textual relevancy over the time

The final user needs back a final list of the most important terms (with the higher tfidf score) for each different discrete time series value. This type of information will be represented in a .json file, each entry will have a different datetime value and it will contain an array for the most relevant terms. Since

all the operations made are on the context of the emails, in this case also all the methods to handle this task are integrated inside the nlpProc.py module.

4.3 Run-time

After the server terminate all his operations at the initialization phase of the system, the client will basically have the possession of two different datasets: directed graph network and undirected graph network. Each dataset contains two lists: nodes and edges. At this point, users can start apply all the run-time procedures needed to elaborate and generate information about the elements and objects visualized by the application. All these operations are handled at the client side, therefore the client is the only responsible part for the correct execution, and the server resources will not take in charge any additional computational activity over the initialization operations already described in the previous section.

4.3.1 Email mining

The email mining operations at this stage are much lighter in terms of computational work, therefore executing them will not compromise drastically the system performance. The operations handled are: general graph metrics/statistics and individual information analysis about the network graph elements. The email mining procedures made at run time, could be done on two different scenarios, and using two different data representation: all the original data or the filtered data (built according to the filtering applications).

Original data

The original data are the basis data representation without any filter adoption on them. As soon as the client retrieve the .json dataset of the data, he can already infer and compute some fixed information, which are

independent from the user interaction with the application and the further filtering operations made.

The new values inferred will be integrated as a new attribute to the original element object. This set of additional attributes are very feasible, and we can add a new attribute for each additional information we mine. Computing the information this way, will relieve a lot of future computational elaborations, since this operations will be done only one time only. Each time we want to get a particular knowledge from the original data, we can get it from one of his attributes. The fixed information that will be generated at this phase are:

Domain: the hostname, such like a list of dot-separated DNS labels

Strength: or the sum of all edges weight connected on a specific node

Degree: the 'in' and 'out' degree will also be calculated in case of directed graph

Emails: a list of all emails where a specific node is involved in

Weight: this attribute will be integrated only to the edges

Filtered data

The filtered data are strictly dependent on the user interaction and the filtering operations chosen. Therefore we can't calculate the information/attributes of each node one time only, when retrieving the dataset after the initialization phase (like the original data occasion). So the related info for the nodes and edges in this case will be calculated each time the user applies a new filtering option or modifies a filter field. The actual implementation and filtering fields will be described in the next Section. 4.3.2. Here we will list the information that will be generated dynamically each time we apply a new filtering option, and therefore update the filtered data collection:

Strength: re-adapting and inserting a new attribute for the node strength in relation to the filtered data.

Degree: re-calculating the degree value (in/out eventually) in relation to the filtered data.

Weight: re-calculating the weight value of each edge in relation to the filtered data.

Subjects: the list of subjects the node or edge is involved in.

Concept: the concept cluster affiliated to the node or edge.

4.3.2 Data filtering

Filter the graph network data (nodes and edges) according to values presented in their attributes. We will let users operate filtering operations through interactive user interface and filter the network in real-time. Filtering options let users focus their searching on specific fields, and remove irrelevant elements. All these operations will be applied on the original data generated from the server, and therefore produce a new set of data to visualize and infer new information from. Data filtering operations will be computed by the client on the .json data previously retrieved from the initialization phase, so their will be no need for any server request in order to fulfill the filtering operations, and thus all handled from the .javascript modules.

Filters

The web application guarantees five basic filtering options:

Node strength: the minimum strength of the nodes that will compose the graph network, the possible values range go from 1 to the maximum possible node strength, users can select this value through an input slider.

Edge weight: the minimum weight of the edges that will compose the graph network, the possible values range go from 1 to the maximum available edge weight, users can select this value through an input slider.

Contact name: any alphanumeric textual expression could be used, if a contact name contain such word it's correspondent node will be included. Users can write the text inside an input search box.

Content: users have the ability to type any textual content, all the messages that contain such text will be included (and therefore also the nodes and edges that are involved with such messages). The textual content could be written inside an input text box.

Time: the values range goes from the date of the first message (sent or received) to the date of the last message (sent or received). users can select a range value through a double input slider.

Implementation

Since all the filtering operations are made at run-time execution, the user dataset at that point will correspond to the one generated by the server in the initializing phase and further sent to the client. The data archive in client possession at this point contains two sub-datasets: for the directed, and undirected graph network (see Section 4.2). The filtering operations made on the filtering fields of 4.3.2 will affect both these datasets. So in case the user readopt a filter value, the corresponding variable will change accordingly, and the system will proceed in applying the changes and rebuild the visualization of both the network graphs. The scheme in Figure.4.4 shows the implementation procedure described. The filtering process (operations inside the funnel of Figure. 4.4) for both the networks will basically follow these steps:

1. Create a filtered dataset copy analogous to the original one.

2. Iterate and retrieve every edge object from the edges list.

3. For each edge object check all the filters (introduced in the previous section).

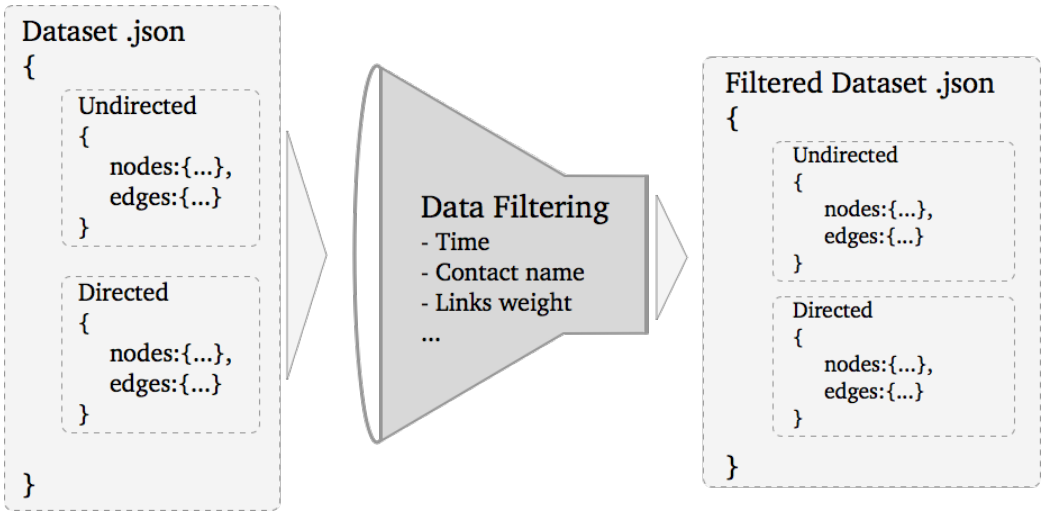


Figure 4.4: Filtering elaboration phases: data conversion to filtered form

4. If the edge fulfills all the filtering requests, it will be inserted in the filtered dataset edges list.

5. Build the list of nodes according to those that appear in the edges.

4.4 Graphical visualizations

The final usable application is web-based, so we have a basic server-client interaction and the client can run the application on a web browser. The application was tested and usable through the major defused web browsers. Web applications use web documents written in a standard format such as HTML, JavaScript and CSS for page styling. All the interactive graphical and visual effects can be made through the definition of some javaScript procedures along with the elaboration of the HTML canvas objects. The engine and libraries used to create a dynamic web content page are: vis.js[2], and d3.js[1]. In this section we will describe the main graphical elements and their dynamics. In the last section we will show a summary graphical view of all the system with the integration of all components, along with a user interaction example with the system.

4.4.1 The network graph

Network graphs are used to represent entities communication, data organization, computational devices, the flow of computation, etc. For our project we used the graph network to generate a social map for the contacts inside the emails archive analyzed. The graph visualization will be adopted in two of the four basic visualization panels: Relationships network and message traffic network (see Section. 3.2.1). Both these visualizations need a dataset composed from a list of nodes and edges, these components along with their attributes define the skeleton composition of a graph. The style and shape of the graph nodes and edges need to be adaptable to dynamic modifications, this feature is needed in order to represent some entities characteristics, e.g: nodes degree, edges weight, nodes domain ...etc.

Graphs will be represented visually by drawing circles for every different vertex (node/contact), and drawing an arc between two vertices if they are connected by an edge. In case the graph is directed, the direction is indicated by drawing an arrow arc. Vertexes and arcs color and size will represent different information according to the user choices. The correct positioning of the vertexes is very important for a conceptual understanding of the network, and for an easier graphical interaction.

We define 4 different style variables to display and view different information: Node size, Edge size, Node color, Edge color. The user can choose what kind of information each one of these variables will show (e.g: node sizes can represent the degree of the node). Each network graph (Directed and Undirected) can employ a different style settings. Here we mention the type of info each variable can forward:

Node size:

- Sum of Edges weight
- Degree
- In degree (Directed case)
- Out degree (Directed case)

- Number messages sent (Directed case)
- Number messages received (Directed case)
- None: all nodes will have a default standard size

Edge size:

- Weight: number of messages sent between the two nodes
- None: all edges will have a default standard size

Node color:

- Cluster: each node will be colored according to the concept cluster it belongs to
- Domain: the email contact domain
- None: all nodes will have same color

Edge color:

- Cluster: each edge will be colored according to the concept cluster it belongs to
- None: all nodes will have same color

To implement this visualization and these behaviors we used a javascript library: vis.js. Vis.js is a dynamic, browser based visualization library. It's designed to be easy to use, to handle large amounts of dynamic data, and to enable manipulation and interaction with the data. The library consists of the components DataSet, Timeline, Network, Graph2d and Graph3d. In order to build our graph we will use the Network component. This visualization is easy to use and supports custom shapes, styles, colors, sizes, images, ...etc. The network visualization works smooth on any modern browser for up to a few thousand nodes and edges. To handle a larger amount of nodes, Network has clustering support. The rendering operations uses HTML canvas.

A vis.Network object needs three basic components for it's initialization: an HTML container, the data (nodes and edges), and an object which defines the options and configurations of the network. The data object used is a Dataset element, this type of data object help us deal with dynamic data, and allows manipulation of the values. Changes made in the Dataset will automatically be reflected and change the view. A DataSet can be used to store any JSON object by unique id's. Objects can be added, updated and removed from the DataSet, and one can subscribe to changes in the DataSet. The data in the DataSet can be filtered and ordered, and fields (like dates) can be converted to a specific type. Data can be normalized when appending it to the DataSet as well. The possible interaction events with the graph will be associated to the Network graph just created, by defining the procedures of the type of events we want to handle. (e.g: when clicking on the graph $on('click')$). The system will create two different vis.Network objects: directed and the undirected representation. The scheme in Figure. 4.5 summarizes what has been said. Some of the possible events and available interactions are:

Hover/blur on nodes: this operation will highlight the contact and all the connected edges.

Clicking on a node or edge: will open a section with all the related information, along with same visual effects of the Hover/Blur operation.

Double clicking on a node: will regenerate the network including only the selected node, the connected edges and his neighbor nodes.

Dragging nodes: dragging and re-positioning the nodes freely according to the user necessities.

4.4.2 Circle packing graphic

The concepts will be represented inside a hierarchical circle graphic representation. The big circles are the different concepts inferred, while the white

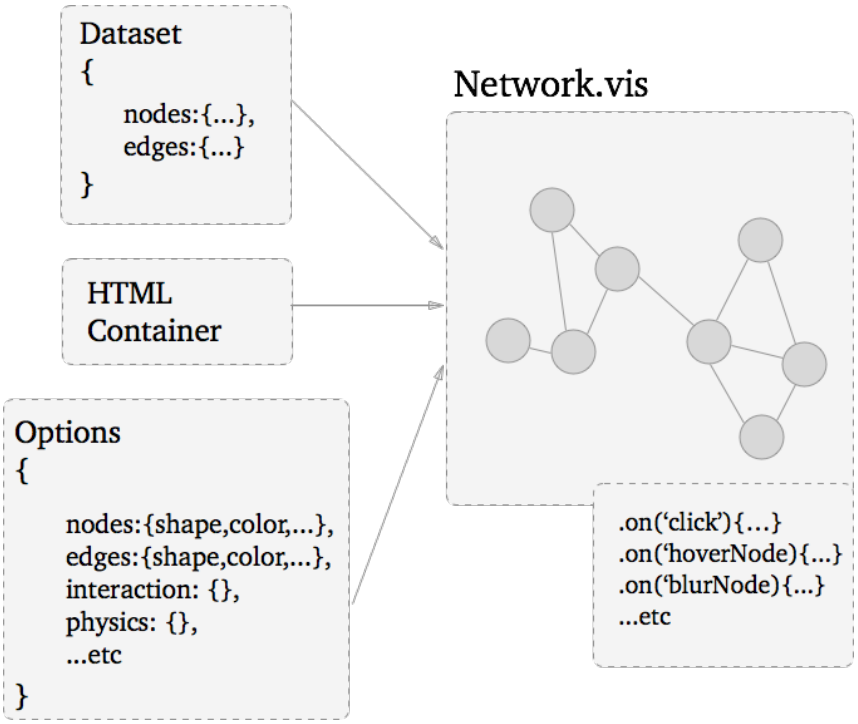


Figure 4.5: Creation scheme of vis.Network object

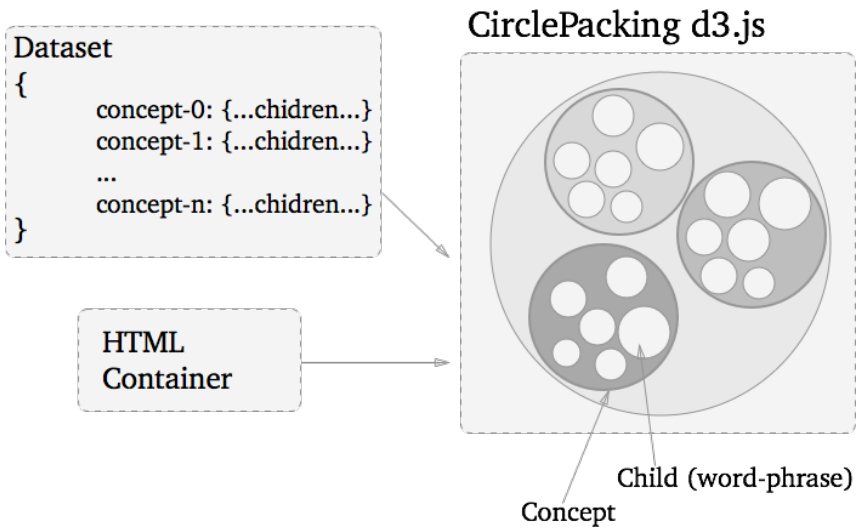


Figure 4.6: Creation scheme of a circlePacking object from d3.js

inner circles represents the words included in each circle. To create this view we used the d3.js javascript library. Like vis.js, it's also used for producing dynamic, interactive data visualizations in web browsers. It makes use of the widely popular SVG, HTML5, and CSS standards to create an embedded graphical object. Users still have the possibility to interact with the web application through mouse events, which can be defined with ad-hoc procedures. In this occasion the only possible event handled is the click on the circle nodes. The input data could be in various formats, although the most common format is JSON, and it's the one we are going to use. In addition to the data, we need to give as an input the HTML basis container where the d3.js object will be generated. The scheme in Figure. 4.6 summarizes what has been said so far.

4.4.3 Timeline graphic

The Timeline graphic is an interactive 2D chart to visualize the data as a function of time. The data items can take place on a single date, or have a start and end date (a range). The view offers several interactive usage with the graphic, such like moving and zooming in the timeline by dragging

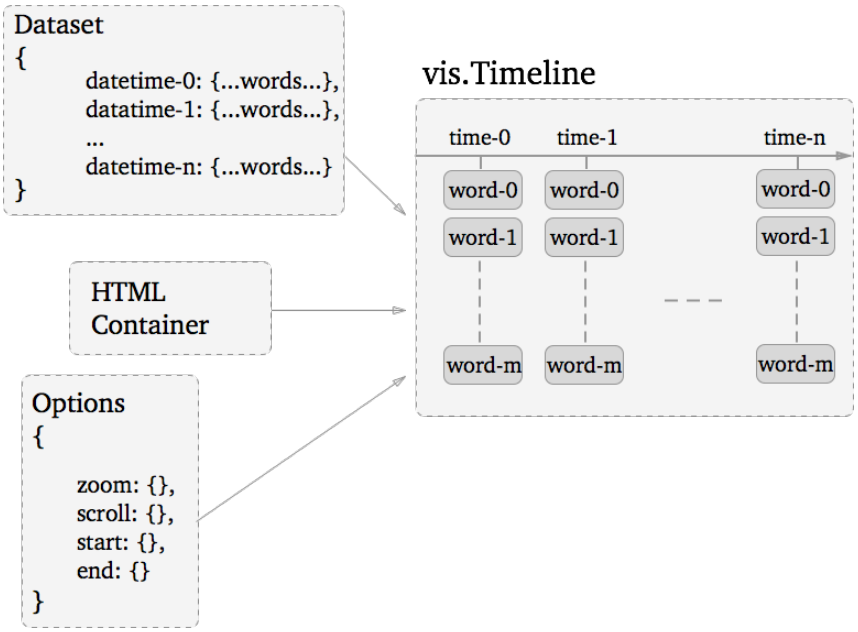


Figure 4.7: Creation scheme of the vis.Timeline object

and scrolling the mouse. Items can be created, edited, and deleted in the timeline, Although in our case the items will be statically created only one time at the initialization of the graphic. The time scale on the axis is adjusted automatically, this operation will also be ignored since the values on the time axis will be represented only as discrete time values. We need to give as an input the HTML basis container, along with the visualization configuration options, such like styling and interactive behaviors. The scheme in Figure. 4.6 summarizes this building procedure.

The possible events handled in this visualization are: the timeline scrolling and the words selections. Mouse clicking on a visualized item (word) will generate the section of information and data related to that item. While scrolling through the timeline will let us visualize all the discrete time values and the correlated words, arranged as a column in relation to their importance (the TFIDF score).

4.4.4 Framework GUI

The framework is a Web application and could be consulted through the major important web browsers at the address: `http://smartdata.cs.unibo.it/emailAnalytics/`. Figure 4.8 shows how the system looks like when the initial loading is done. The previous described graphic visualizations: network graphs, circle packing view, and the timeline graphic. Could be switched through the item(4) of Figure. 4.8, which also shows the default network graph panel view. In Figure. 4.9 we show the aspect of the other two visualizations in the framework. Here we give a description of the parts illustrated in Figure. 4.8:

1. The filtering options (described in section 4.3.2)
2. This button will open the visualization options we described in 4.4.1
3. Will open a help information window to describe the possible interactions with the current visualization
4. A set of 4 tabs to switch from a view to another: network graphs, circle packing view and the timeline graphic.
5. A slider to set the time range (filtering option)
6. The two info menu layers which will generate the related info on the wanted elements.
7. A panel to show a list of all the information according to the menu selections we made in (6)

4.5 Comparison with other tools

Previously we talked about some important email forensic tools already diffused and used in literature (see Section. 2.5). The comparison between the tools was made on different criteria and by specifically pointing out different

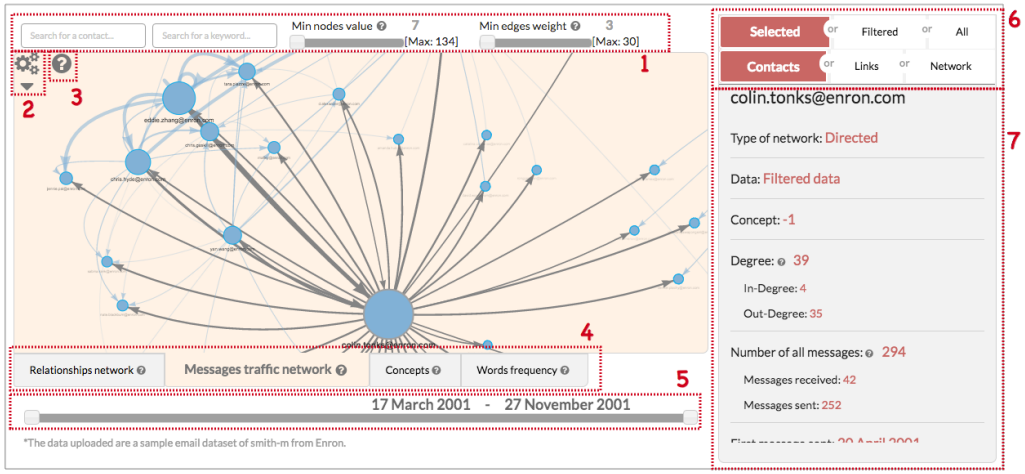


Figure 4.8: Framework GUI: (1) Filters, (2) View options, (3) Help info, (4) Panel tabs, (5) Time filter, (6) Info menus, (7) Info section

sub matters. In this section we will try classify where our framework stands compared to all the others.

The Table. 2.4 of the previous chapter includes all the tools and their features, we will bring up again the content of the table, and give a conclusive percentage summary for each criteria, and compare such result with the actual features included in our framework. Table. 4.2 shows this representation.

Features	Other tools (from Table. 2.4)	Our framework
1) Operating system	Windows, Linux, Web App	Web App
2) Search/filter options		
2.1) Words in context	Yes: 7/8, No: 1/8	Yes
2.2) Contact name	Yes: 8/8	Yes
2.4) Sending time	Yes: 8/8	Yes
2.5) Filtering in a time range	Yes: 5/8, No: 3/8	Yes
2.6) Subjects/threads name	Yes: 6/8, No: 2/8	Yes
2.7) Contacts relevance	Yes: 2/8, No: 6/8	Yes
2.8) Relations relevance	Yes: 2/8, No: 6/8	Yes
2.9) Concepts/topics affinity	Yes: 1/8, No: 7/8	Yes
2.10) Contacts relations number	Yes: 5/8, No: 3/8	Yes
2.11) Filtering/searching combination	No: 8/8	Yes
3) Information provided		
3.1) Messages traffic information	Yes: 8/8	Yes
3.2) General SN stats and metrics	Yes: 2/8, No: 6/8	Yes
3.3) Contacts and relations (SN)	Yes: 2/8, No: 6/8	Yes
3.4) Documents/Attachments analysis	Yes: 6/8, No: 2/8	No
3.5) Calendar data analysis	Yes: 6/8, No: 2/8	No
3.6) Contacts and relations relevance	Yes: 2/8, No: 6/8	Yes
3.7) Sending and receiving messages streams	Yes: 2/8, No: 6/8	Yes
3.8) Retrieving original documents	Yes: 7/8, No: 1/8	Yes
3.9) Keywords occurrences in context	Yes: 1/8, No: 7/8	Yes
3.10) Geolocation	Yes: 4/8, No: 4/8	No
3.11) Semantic analysis of the context	No: 8/8	Yes
3.12) Urls/links detection in context	Yes: 6/8, No: 2/8	Yes
3.13) emails detection in context	Yes: 5/8, No: 3/8	Yes
3.14) Temporal occurrences detection	No: 8/8	No
3.15) Word phrases detection	No: 8/8	Yes
3.16) Words relevancy ranking	No: 8/8	Yes
3.17) Concepts/topics auto detection	No: 8/8	Yes
4) Supported email formats	PST, OST, MBOX etc	MBOX
5) Visualization method		
5.1) Network graph	Yes: 2/8, No: 6/8	Yes
5.2) Charts and bars	Yes: 6/8, No: 2/8	Yes
5.3) Structured lists	Yes: 8/8	Yes
5.4) Geographic map	Yes: 3/8, No: 5/8	No
5.5) Cluster map	Yes: 1/8, No: 7/8	Yes
5.6) Dynamic interaction	Yes: 2/8, No: 6/8	Yes
5.7) User-friendly interface	High: 4/8, Medium: 2/8, Low: 2/8	High
6) Export format	PDF, HTML, CSV etc	N.N
7) Software licence	Commercial, Open source	Open source

Table 4.2: Our framework features compared to the features included in the frameworks of Table. 2.4

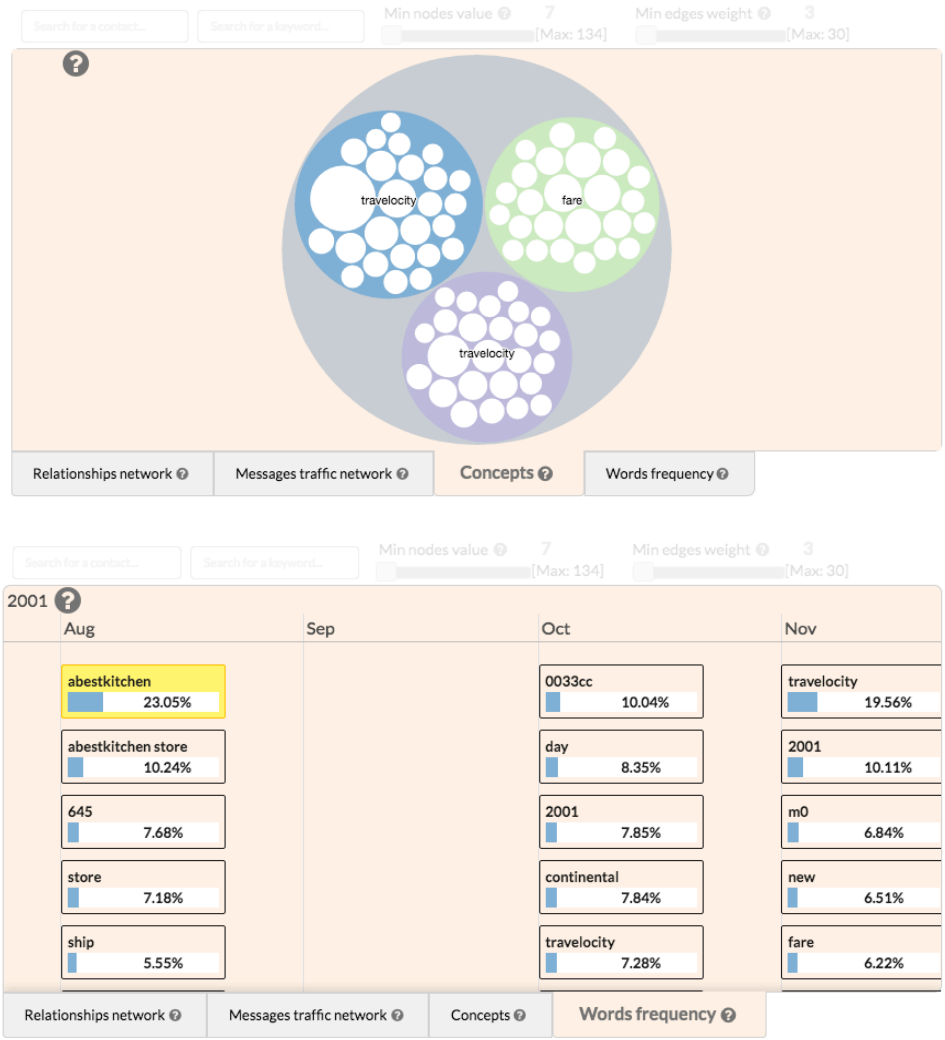


Figure 4.9: (A) The circle packing graphic for concept classification (B) The timeline graphic of the word phrases relevancy over time

Chapter 5

Evaluation: a case of study

To test and correctly evaluate the framework we need a dataset we can rely on, a very popular and trustworthy archive of emails we can use is the 'Enron' dataset, which has been frequently used for scientific experiments. In the first section of this chapter, we will talk about this choice and why we picked specifically this dataset, by also giving a general and juridical background on Enron and the more famous 'Enron case'.

The Enron dataset will be used in all our tests, and we will process the evaluation in two phases:

General framework features evaluation: in this phase the main purpose is to test the features and elaborate the results given by the framework. The results obtained hides interesting aspects, which we will try to infer and point out. This evaluation test will be made separately (in two different sections), for two fundamental aspects: the social network generation, and the textual content analysis. In this case the experimental data is randomly selected from the dataset. In some occasions we will compare the results obtained with other past related works, as a possible methodology to give more relevancy and significance to our testing conclusions.

Forensic investigation: use the framework with a precise objective, in this case we will take the 'Enron scandal' as a case of study. We will ap-

proach the dataset from an investigative eye, aiming the most relevant data and persons who were mentioned in the juridical reports. We will compare and try to find analogies between the information we discover and the actual facts reported.

5.1 The Enron case

Enron corporation was formed in 1985 under the direction of Kenneth Lay, who became the CEO of it for most of its existence. Along with Mr. Lay, also president and chief operating officer Jeffrey Skilling took over the position of chief executive for one year in 2000-2001. Enron was established through the merger of Houston Natural Gas, a utility company, and Internorth of Omaha, a gas pipeline company. The company was based in Houston, Texas. Within 15 years Enron became the nation's seventh-biggest company in revenue by buying electricity from generators and selling it to consumers. At the end of 2001, the financial condition of Enron was reported as institutionalized, systematic, and creatively planned accounting fraud, this fact was known since that as the 'Enron scandal'.

One of the bigger reasons is the fact that Enron officials began to separate losses from equity and derivate trades into "special purpose entities"(SPE); partnerships that were excluded from the company's net income reports. This led to a systematic omission of negative balance sheets and income statements from SPE's in Enron's reports, which led to an off balance sheet financing system.

Further interesting studies and legal actions were conducted, to reveal the main people responsible for the definitive bankrupt. On January 2006, the New york times made an article listing 10 of the major figures involved in this scandal [4], along with the 2 responsible CEO. These characters played different roles, some have admitted to helping artificially increase profits and hide losses and debts, while others tried to blow the whistle on the deceptions. We will briefly mention these figures, and their roles. (for further details the

related article is [4]).

Kenneth Lay He joined Houston Natural Gas Co. as chairman and CEO in 1984. The company merged with InterNorth in 1985, and was later renamed Enron Corp. In 1986, Kenneth Lay was appointed chairman and chief executive officer of Enron. In 2001, Lay sold large amounts of Enron stock in September and October as its share price fell. All told, he liquidated more than $300 million in Enron stock from 1989 to 2001.

Jeffrey Skilling in 1990, Skilling was hired away from McKinsey by Kenneth Lay to work at Enron Corporation. Skilling was named chairman and chief executive officer of Enron Finance Corporation and became the chairman of Enron Gas Services Company. He was named CEO of Enron, replacing Lay, in 2001. In August 2001, amidst the California energy crises, Skilling unexpectedly resigned and sold almost $60 million in Enron shares. He mentioned that the reasons for his resignation is due personal factors.

Andrew S. Fastow Enron's financial chief, is considered the main character behind the off-balance-sheet special purpose entities.

Ben F. Glisan Jr. He became part of the inner circle and helped conceive and execute several financing schemes that hid company losses.

Mark E. Koenig The director of investor relations at Enron, he was managing some suspicious calls.

Lou Lung Pai He headed several divisions at Enron, including Enron Energy Services. His name appears on a list of potential witnesses for the defense in the trial of Mr. Lay and Mr. Skilling.

Kenneth D. Rice He held several posts during his 20-year career at Enron, including chief executive of its high-speed Internet unit. He was a favorite of Mr. Skilling, accompanying him on several trips.

Greg Whalley Enron's former president, once created a hypothetical futures contract for Popsicles. He has cooperated with investigators, but the legal cloud over him led a Swiss bank, UBS, to let him go shortly.

Nancy Temple An Andersen Lawyer. The jury hearing the criminal case against Andersen focused on advice that Ms. Temple, gave to Andersen's lead partner on the Enron account.

Rebecca Mark she was an Enron ambassador abroad. She cooperated with a Senate committee that investigated Enron improprieties in international deals.

Sherron S. Watkins Sherron S. Watkins is remembered for the letter she wrote as a company vice president in August 2001 to Mr. Lay, describing improper accounting practices at Enron. Months later, Enron collapsed.

Vincent J. Kaminski He was Enron's managing director for research. For months before Enron's demise, Vincent J. Kaminski warned superiors that the off-the-books partnerships and side deals engineered by Mr. Fastow were unethical and could bring down the company.

5.1.1 The dataset

Since our framework is basically based on analyzing private email collections, finding and working with a real dataset is a challenging request. This is due to privacy concerns, since using such datasets treat sensitive and private aspects of the people involved. Email datasets belonging to companies and organizations, are a good example of private collections of data that operate under the same domain (the company).

Enron email archive dataset, is a unique large dataset which contains more than 2000 emails. When Enron collapsed in 2001, all these emails were made public, and a lot of publications based their analysis on these data. The dataset used for the results analysis, is selected from two years of time

span between January 2000 and December 2001 as the email collaborations in this period of time look most realistic, and we have public data about many Enron affiliated people. It contains data from about 150 users, mostly senior management of Enron, organized into folders.

This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation. Further re-elaboration of this dataset was made (basically to remove sensitive private data). The archive we are using was downloaded from the Carnegie Mellon University School of Computer Science [39]. The dataset does not include attachments, and some emails have been deleted "as part of a redaction effort due to requests from affected employees". Invalid email addresses were converted to something of the form user@enron.com.

If we take in consideration the main characters mentioned on the previous section, then our dataset contains the personal email archives of: Kenneth Lay, Jeffrey K. Skilling, Greg Whalley, and Vincent J. Kaminski. Since these people are considered key actors of the 'Enron scandal', it might be useful giving them a further special attention.

Data format

The available dataset contains emails from about 150 Enron member, and it's organized into folders, very similar to a .maildir email format. Each folder will represent a different contact, and will contain it's own internal folder organization. Since our system take as input only .mbox files, we converted the current original format to .mbox through a python module. Each user will be represented in a separated .mbox file. Our system might take only one or multiple files as input.

5.2 Social network analysis

Our social network analysis focuses on the relations among and between the entities of the analyzed archive. Since we are using the Enron dataset,

the Enron users are the entities. We can take as input any personal email archive of any Enron member, or combine multiple archives together in order to analyze a larger amount of data, and try elaborate the possible correlated information.

The evaluation will consist in two types of experiments: by applying as input separated archives of 3 randomly selected Enron users, and by combining all the 3 archives. This kind of research is of an explorative nature, and aim to understand the social networks analysis provided by our framework. We will try to study the results and infer relevant knowledge and social behaviors with the other contacts in the network.

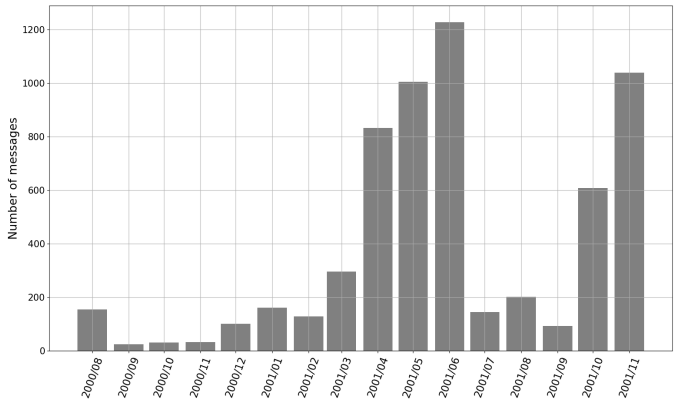
The analysis will treat two main sub fields of study: the messages traffic and the contacts relationships/communities. In the first case the graph network generated is a directed one, so we will have the opportunity to distinguish the sending and receiving actions, along with the archive owner messaging activities. The second graph network is an undirected one, this will emphasize the search on the communities and group of contacts associated to common subjects or groups of work.

5.2.1 Individual users

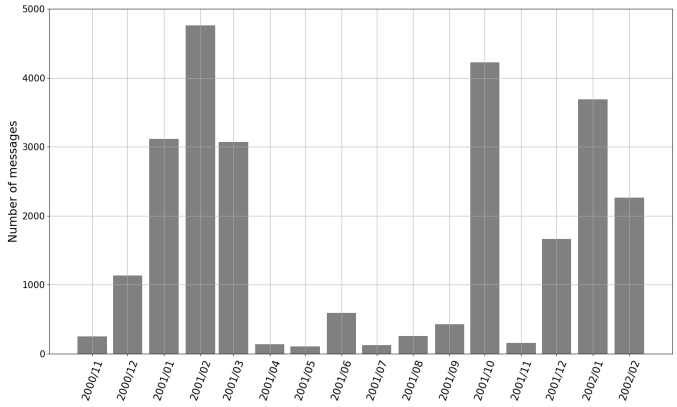
In this stage we will achieve a social network analysis when giving as input individual Enron users email archives. We will select the individual archive of 3 randomly chosen characters: Smith, White and Ybarbo.

From this analyzes we might obtain different information. Our object is to try evaluate such information and infer relevant social behaviors and knowledge about the single contact actions and stats.

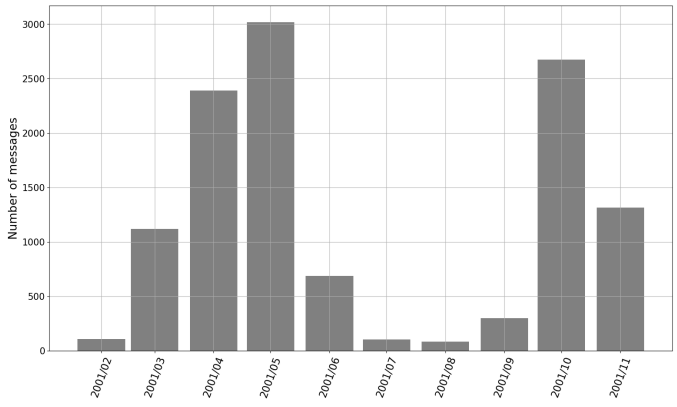
Let's first take a look at the messages traffic distribution over the time, Figure. 5.6 shows the number of messages made by all the addresses in the contacts list of each emails collection, so each sub-figure represent a different Enron archive: Smith, White, and Ybarbo. As we can see from the previous Figure. 5.6 the email-archive of White (see Figure. 5.1b) is the one with the highest number of messages, particularly we can notice a high message traffic



(a)



(b)



(c)

Figure 5.1: Network messages traffic for (a)Smith /(b)White /(c)Ybarbo archives

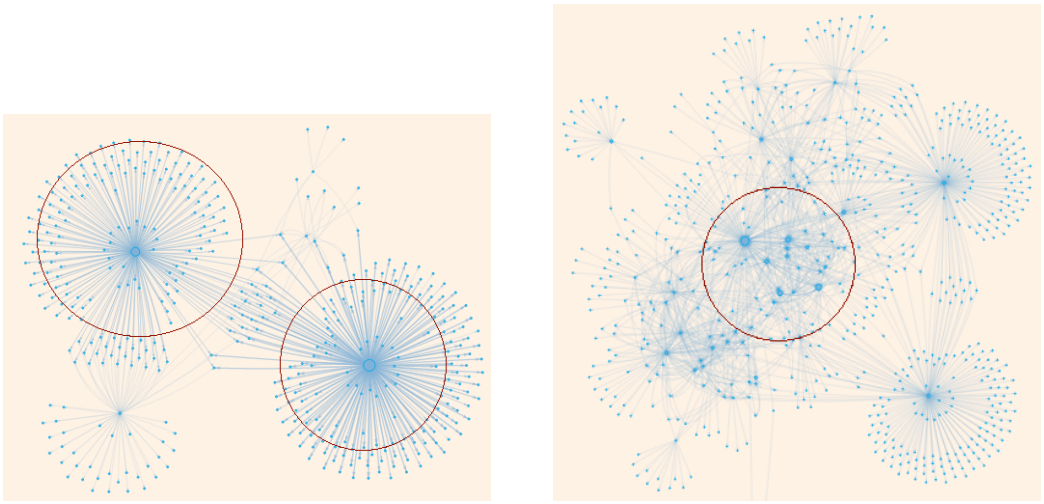
activity on 'February 2001' and 'October 2001', so it might be interesting focusing our search on these dates and give an explanation to this.

Concentrating our search on these two dates will generate the graph network of Figure. 5.2. From 'February 2001' we can clearly see 2 different group of nodes having two nodes that generate a high quantity of out edges. If we select these contacts (rhonda.denton@enron.com and cheryl.johnson@enron.com) and look at the list of email-subjects, we notice the fact that they address a lot of contacts in almost all the emails that they send. This can make us think that these users are very interested in spreading information to a lot of contacts at same time, and they work as important hubs to all Enron users. Reasons could involve a common project, or informative news useful to a big group of users.

The second date 'October 2001' will generate the graph network of Figure. 5.2b. This time we are not noticing any remarkable group of nodes. Although, as we also point it out in the Figure, some nodes look much stronger (bigger), this might suggest to restrict our analyzes on that window. Looking at the main nodes composing the cluster and on the type of email subjects they send and receive, we see a frequent common title in the form of: "ERV Notification: ... Report By Trader... ". From this we can easily deduce the fact that the message traffic between these nodes is highly influenced by notification or report subjects. This final result might get more interesting further on, when visualizing White's archive from a different perspective, involving the relationships and working groups.

Another interesting investigation could involve monitoring and separating the sending and receiving operations made by the users. In Figure. 5.3 we show the results obtained from this point of view, when selecting the email owners of each archive analyzed. Each one of these sub-graphic brings different conclusions:

Smith Figure. 5.3a underlines how Smith used to have more message sending activities rather than receiving, specially for the first months of the 2001. If we concentrate our timeline filtering field on these values we



(a) (b)

Figure 5.2: White's graph network structure for two different scenarios: (a) February 2001, (b) October 2001

can clearly see that the resulting network for the node 'Smith' will look like a star network, all the edges will get out from the smith node and reach the other neighbors. So in that occasion Smith acted like a hub for the others and was spreading messages info common to a particular group of nodes (the neighbors).

White for White case we notice a relatively low number of messages that involves him directly, if compared with the total number of messages exchanged in the whole network (see Figure. 5.1b). This means we will have a lot of additional data (messages) that will help us build a vast network with nodes not directly connected to White. In addition we have a balanced situations between sent/received msgs for almost all the months, as a result we might consider White messaging activities ordinary.

Ybarbo Also in this occasion, like the White case, we have a significant low number of messages in relation to the total. Particularly at the end of

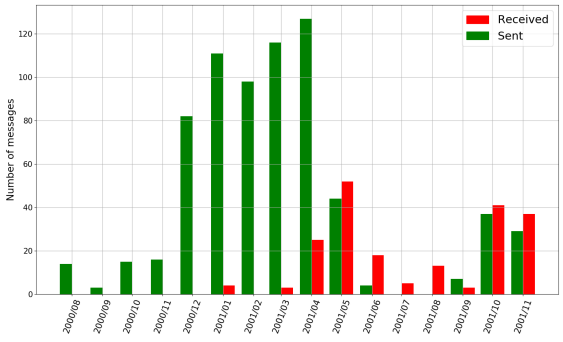
the 2001 year we have a significant decrease in the number of messages sent. If we focus our search on that period of time we found out emails were Ybarbo acted more as a listener, and has never sent any reply to the subjects were he got involved. A further textual analysis of these emails might give us more clues on this behavior.

Communities

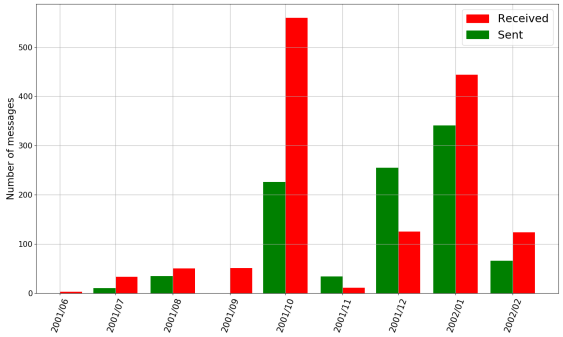
All the previous analysis take in consideration the email archive owners and the directions of the messages flow. An investigator might want to bring to light the possible communities and group of contacts cooperating on common subjects without giving any relevancy on the flow of messages direction.

Let's apply this kind of analysis on the previous email archives of Smith,White and Ybarbo, and see the results obtained. We will try elaborate these results and see if we can discover relevant information.

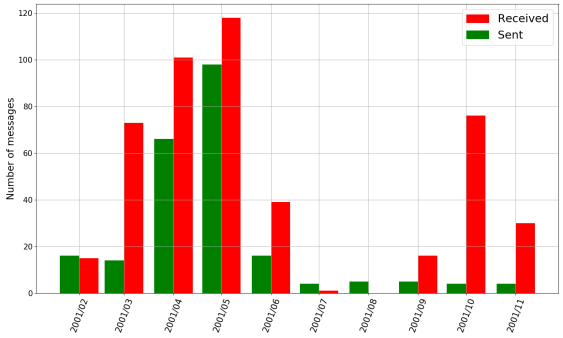
White If we take a look at the 'Relationship network' of White, the network shape will look like the one we have on Figure. 5.4. Taking a look at Figure. 5.4a we can see the results based on a minimum edge weight value of 1, and therefore as we can see it's difficult to distinguish the stronger cooperation clusters from all the others. A good solution is to increment the edge weight and rebuild the visualization. In Figure. 5.4b the minimum edge weight is set to 10, and the results are certainly more clear. Here we distinguish 3 different groups, with different structure shape. As we can see we clearly have a star network sub-network, star networks consists of one central node, which typically acts as a hub, and transmit messages to his connections. If we take a look to the set of emails exchanged in that star network, then what comes to the light is having a large set of emails with the word 'Notification' in there subject. This puts into effect what has been told on the behavior of the central node of a star network.



(a)

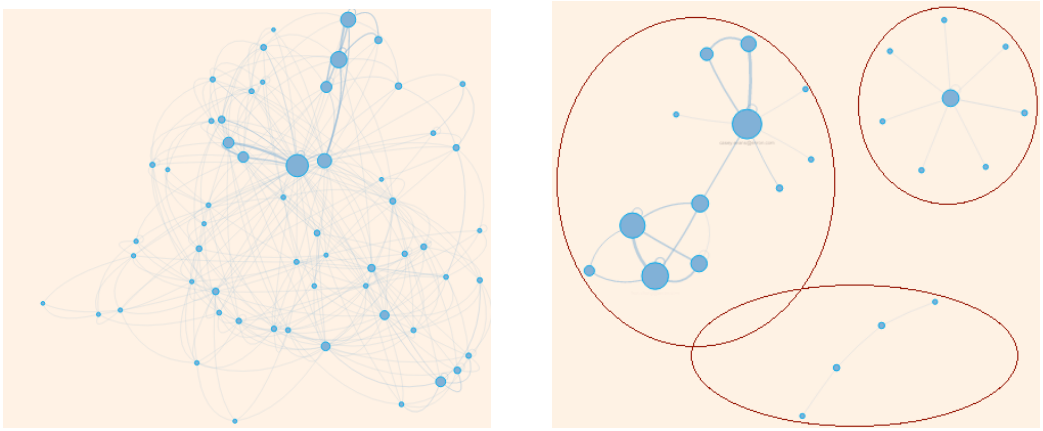


(b)



(c)

Figure 5.3: Sending/receiving messages traffic for: (a)Smith (b)White (c)Ybarbo archives



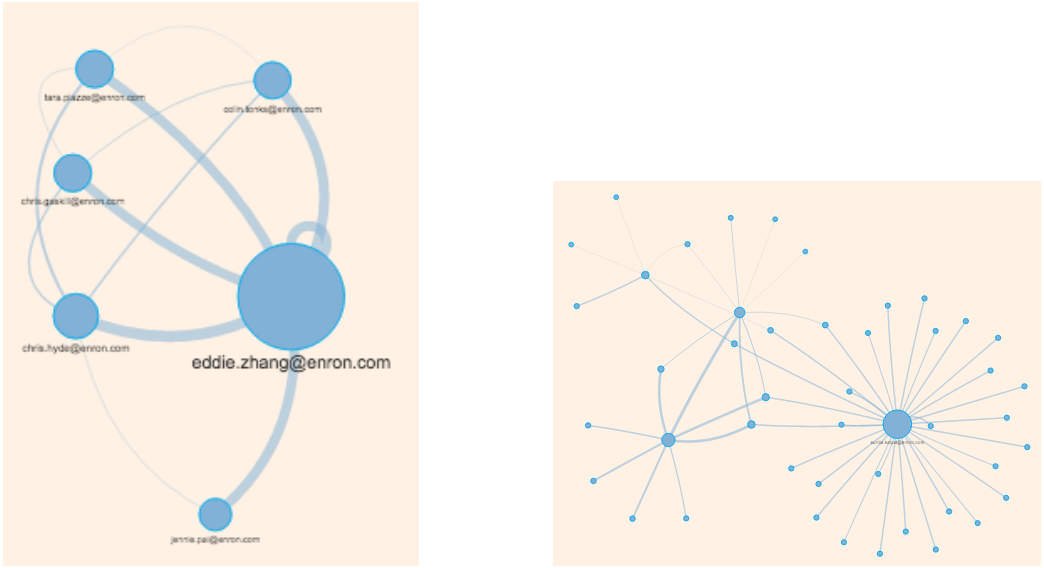
(a) (b)

Figure 5.4: White's relationships graph network, with minimum edge weight: (a)1, (b)10

Another sub-network structure we have is a partially connected network. This type of formation will mostly appear when the users involved cooperate or work together on a common subject. Nodes size reflects the importance of individuals inside a working group. Selecting nodes from such group reveal a possible common working project named "Power West". Correlated emails where exchanged for notifications and reports on that work.

Ybarbo The network graph generated in this occasion appears to be a combination of different star networks (3 at least, see Figure. 5.5b). The stronger node of this network is 'sunita.katyal@enron.com' which appears to be a high host and source of messages. The subjects treated by him are mostly related to weekly updates and reports.

Smith In this case the number of emails in the archive is much fewer in respect to the previous users archives analyzed. This will decrease the overall relevance of the system results. It's therefore expected to see few user clusters (see Figure. 5.5a). The only connected group we obtained have a central important contact "eddie.zhang@enron.com". The focus



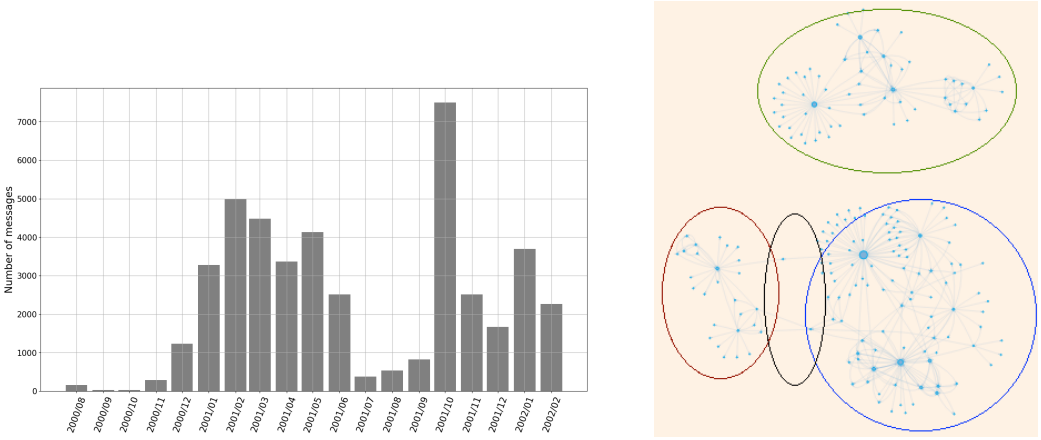
(a) (b)

Figure 5.5: The relationships graph network of: (a)Smith (b)Ybarbo

of this contact was basically on a particular subject: "west pipeline". It appears that such emails were related to a server error and technical problems while working on a project.

5.2.2 Multiple archives

In this section we will combine all the previous three email archives (Smith, White and Ybarbo), in a one larger dataset. This operation will let us emphasize our analysis on common contacts and relations between different users. Since we are using Enron as the basis of all the databases uploaded, it's expected to see some common accounts that act like bridges between the different users. In Figure. 5.6b we show an overview of the resulting graph network: the green circle surrounds Ybarbo sub-network, the red circle is the Smith's sub-network, and the blue one is the White sub-network. As we can clearly see Ybarbo sub-network is visibly separated, and got no edges with the other two sub-networks. On the other hand, Smith and White sub-networks are connected by some nodes: the nodes that re-



(a) (b)

Figure 5.6: Messages traffic: (b) graph network, (b) as a function of time. When uploading multiple email archives: Smith, White and Ybarbo.

sides inside the black circle. These nodes act like a bridge between the two networks. These nodes are specially involved in messages receiving events, it's interesting to check what are the subjects that combine White and Smith with these particular nodes. The list of emails retrieved in this occasion can gain more significance if analyzed textually also. The relationships network and communities detection applied to this archive, will give us the results of Figure. 5.7. As we can see, comparing to what we have previously seen, we don't have common users that will enlarge or modify the clusters obtained for each user individually. Therefore the final result shows each group of collaboration separately, and we can distinguish the actual source of these groups. The red circle surrounds Smith's clusters, the green circle surrounds Ybarbo's clusters, while the blue one surrounds White's clusters (see Figure. 5.7).

5.3 Textual mining

Our framework actuate two basic textual analysis: textual relevancy monitoring through the time, and the classification of different topics/concepts.

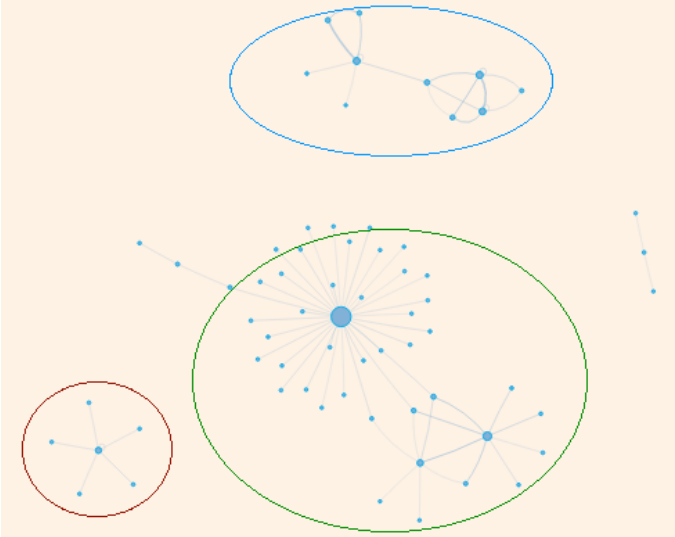


Figure 5.7: The relationships graph network for a multiple email archives as input

In this section both these aspects will be treated, in two different sub-sections.

5.3.1 Terms relevancy over time

This type of analysis will let us retrieve a graphical representation for the most representative terms as a function of discrete time intervals. Since the larger possible time span between two dates in any uploaded archive (or archives combination) for Enron could be 2 years, the system will list the terms with a granularity of a month (will list the most important terms for each different month), The number of representative terms to visualize for each month is 15, these terms will be ordered according to their TFIDF score. Note that the final terms produced might also be the combination of multiple words (maximum 3) according to the n-gram model (see Sec. 3.2.3).

The tests will be applied again on the same 3 randomly chosen Enron archives of the previous section: Smith, White and Ybarbo. Here we mention the most notable facts and results obtained.

Smith (Table. 5.1): Looking at the final results we can easily point out

several months in which Smith used a lot of terms related to travels and journeys, e.g: September 2000, and October 2000...etc(see October 2000 as example). An interesting month is August 2001, and while looking at some of the most important terms, we can see "judge", "el paso", "wagner", "commission" ...etc. These refers to a more general subject of a "El Paso Corp violating some federal rules ". In April 2001, "dynegy" was a very frequent word in the top of the table, in this month there was already some talks about a project for merging the two companies, Enron and Dynegy.

White (Table. 5.2): On June 2001, we have the words "calendar", "time", and "appointment" very frequent, as it's almost already clear in this month we have some emails of White where he was updating his calendar with new appointments. A notable fact emerges by looking at terms generated in December 2001, January and February 2002, as we can see we have a high number of non relevant terms, which points out the importance of having a good cleaning data process.

Ybarbo (Table. 5.3): Looking at May and June 2001 we can see that we have almost the same terms, mostly focused on "dpc" and "mseb", DPC stands for Dabhol Power Company another gas company affiliated to Enron in India.

October 2000	April 2001	August 2001
nov : 0.139	dynegy : 0.141	el paso : 0.13
continental : 0.12	hou dynegy : 0.14	el : 0.109
flight : 0.094	ngccorp : 0.107	wagner : 0.086
nov arrive : 0.089	hou dynegy ngccorp : 0.106	pipeline : 0.067
ticket : 0.088	hou : 0.087	said : 0.062
arrive : 0.064	2001 : 0.061	judge : 0.062
itinerary : 0.053	april : 0.056	ferc : 0.059
continental airlines : 0.051	priceline : 0.048	merchant : 0.058
depart : 0.05	fares : 0.04	executives : 0.056
receipt : 0.043	apr : 0.039	commission : 0.055
escs : 0.042	please : 0.037	california : 0.054
flight continental : 0.042	april 2001 : 0.036	gas : 0.052
meal service : 0.042	01 : 0.036	wagner said : 0.051
mi economy coach : 0.042	hou dynegy dynegy : 0.034	abestkitchen : 0.05
depart nov arrive : 0.04	new : 0.032	wise : 0.05

Table 5.1: Terms with the highest TFIDF score as a function of time for Smith (Enron email archive)

June 2001	December 2001	February 2002
description : 0.124	december : 0.122	february : 0.123
calendar entry : 0.076	stacey : 0.087	2002 : 0.111
detailed description : 0.076	december 2001 : 0.083	ubs : 0.093
chairperson : 0.074	2001 : 0.078	message : 0.072
time central standard : 0.072	cn : 0.077	sent : 0.066
description calendar entry : 0.069	original : 0.064	original : 0.065
detailed description calendar : 0.069	webster : 0.063	original message : 0.063
standard time chairperson : 0.067	sent : 0.061	ubsw : 0.061
central standard time : 0.065	original message : 0.06	please : 0.059
time : 0.061	message : 0.059	subject : 0.058
standard : 0.05	subject : 0.057	pwr gas : 0.05
calendar entry appointment : 0.049	west : 0.055	stacey : 0.047
appointment description : 0.049	word : 0.046	sent february : 0.046
entry : 0.049	white : 0.044	tuesday february : 0.044
calendar : 0.049	bankruptcy : 0.044	netco : 0.043

Table 5.2: Terms with the highest TFIDF score as a function of time for White (Enron email archive)

May 2001
power : 0.121
dpc : 0.102
mseb : 0.094
may : 0.075
2001 : 0.07
government : 0.065
said : 0.062
state : 0.062
maharashtra : 0.056
dabhol : 0.056
may 2001 : 0.054
project : 0.05
rs : 0.047
centre : 0.043
godbole : 0.043

June 2001
power : 0.14
dpc : 0.109
mseb : 0.099
2001 : 0.077
lenders : 0.073
june : 0.068
dabhol : 0.058
said : 0.057
per : 0.05
state : 0.049
rs : 0.047
foreign : 0.044
project : 0.044
india : 0.043
electricity : 0.043

Table 5.3: Terms with the highest TFIDF score as a function of time for Ybarbo (Enron email archive)

5.3.2 Concepts classification

To evaluate the concepts classification and terms clustering procedures (see Section.3.2.2 for theoretical background), we compared our results with the results of the work made by Decherchi .et al [12]. The main purpose of Decherchi .et al work, was building and testing text clustering procedures for forensic analysis aims. The interesting aspect is the fact that they also used the Enron database as a tool for testing their approach, such that the tests were applied on five Enron users email archives randomly selected. In order to get truthful and meaningful results we decided to take same Enron archives and give those as input to our concept classificator.

The Decherchi .et al approach is based on, a Term-Frequency (TF) process for term weighting, and on elaborating the distance between the vector representation of the documents through a k-mean algorithm to create k clusters of terms. The final results cover the most important terms obtained for 10 different clusters, and the terms considered don't include numbers.

These tests were made on the archives of: Smith, White, Solberg, Ybarbo and Steffes (see Tables. 5.4, 5.5, 5.6, 5.8, 5.11).

In these tables we show our results and right after the results of Decherchi .et al. The order of the clusters don't have any relevancy. The terms collected from our approach consider the combination of multiple words since we used the n-gram models for the construction of the vocabulary. This final aspect will turn out to be very beneficial specially when we need a more detailed term, in order to get a more specific cluster definition. Considering the final results obtained in the tables, we can point out some interesting facts that arise:

Smith (Table. 5.4) As we can see our final clusters are very similar to those of [12], a notable interesting fact is that some clusters relate to private and personla duties, we can see that from clusters: 2,6 and 10, and from [12] clusters: 7 and 10. Another interesting cluster from our results is the 8th, which might be associated with the 4th cluster of [12], Although in this case it's we can see how the n-gram model made our cluster's terms more specific (e.g: "natural gas intelligence" vs "gas").

White (Table. 5.5) If we take a look at cluster 8 of [12] we see it's summarized with only one word 'power', although our corresponding cluster (most similar), the 8th, get's more specific and reveal the context where the word 'power' appears.

Solberg (Table. 5.6) In this case for both tables we have some clusters talking about some 'data errors' that have occurred, although if we look at our 2nd and 3rd cluster we notice that the n-gram representation made it possible distinguish what type of 'data error' happened, e.g: in the 2nd we have terms like "cannot perform operation" or "unknown database alias", and in the 3rd we have "manual intervention required" or "schedule download failed".

Ybarbo (Table. 5.8) These results are those that represents most the problem concerned with including numerical text. As we can see for exam-

ple our 4th cluster include a number that supposedly is a phone or fax number. Not filtering this text during the pre-processing phase cause these numbers to upper, which might reduce the importance of other relevant terms.

Steffes (Table. 5.11) The 1st cluster of [12] is represented by 4 acronyms "FERT","RTO","EPSA" and "NERC". The corresponding cluster that we obtained is cluster 4, but as we can see we have other terms to enrich it, e.g: "call","conference call"... etc. This help us get a more general idea about where these acronyms are used.

Cluster	Ten most relevant words (Our results)
1	fares, fare, ctl, dps1, deals, service, earn, new, cruise, ctl service
2	priceline, request, hotel request, long distance, distance, hotel, car, long, free, click priceline
3	west, report, name west, category, cd, name, 2001, 2001 available viewing, 2001 published 2001, available viewing website
4	msg, error, 2001, dtm msg desc, msg desc status, msg received dtm, name msg received, pipeline name msg, received dtm msg, morning
5	thru, sat 2001, outages, scheduled outages, sat, scheduled, 2001, 713, 2001 ct, 2001 pt
6	sheraton, hilton, day, miles, specials, co travel, co, travel specials, co travel specials, rates
7	smith, matt, david shank, smith matt, original message, sent, original, david, subject, message
8	annually, intelligence, gas, index, intelligencepress, natural gas intelligence, gas price index, natural gas, natural, publications
9	service lang en, en, ctl service lang, airfare, rebates next, bush intercontinental iah, houston bush intercontinental, bush intercontinental, next, travel
10	pep, feedback, hours, receipt, ticket, process, trip id, may, id, trip

Cluster	Most frequent and relevant words ([12] results)
1	employee, business, hotel, Houston, company
2	pipeline, social, database, report, link, data
3	ECT, EnronXg
4	coal, oil, gas, nuke, west, test, happy, business
5	Yahoo, compubank, NGCorp, Dynegi, night, plan
6	shank, trade
7	travel, hotel, continent, airport, flight, Sheraton
8	Questar, Paso, price, gas
9	schedule, London, server, sun, contact, report
10	trip, weekend, plan, ski

Table 5.4: Smith textual clusters, our results compared to [12] results

Cluster	Ten most relevant words
1	report, category, cd, name, west, 2001, position, peak, position report, name west
2	description, time, calendar entry, detailed description, chairperson, central standard time, standard, time central standard, standard time chairperson, calendar
3	chairperson stacey white, stacey white detailed, white detailed description, stacey, time chairperson stacey, stacey white, white, 2001 time central, date 2001 time, 2001 time
4	06 central time, central time us, gmt 06 central, time us canada, central time, gmt, 06, canada, us, 2002 gmt 06
5	peak position report, position report trader, peak position, trader, position report, peak, west peak position, position, report trader category, report trader erv
6	06 central time, central time us, gmt 06 central, time us canada, central time, gmt, canada, 06, 2002 gmt 06, 2002 gmt
7	webster, merriam webster, word, word day, mail, mw wod, request listserv webster, mw, request, via
8	power desk daily, desk daily, power desk, daily, desk, daily position report, desk daily position, position report, report, east power desk
9	east, name east, category name east, east toc hide, name east toc, power east, power peak, name power east, named power east, report name power
10	request, resource, id, common, auth emaillink id, corp srrs auth, itcapps corp srrs, srrs auth emaillink, approval, act upon request

Cluster	Most frequent and relevant words ([12] results)
1	meet, chairperson, Oslo, invit, standard, smoke
2	confidential, attach, power, internet, copy
3	West, ECT, meet, gas
4	gopusa, power, report, risk, inform, managment
5	webster, listserv, subscribe, htm, blank, merriam
6	report, erv, asp, EFCT, power, hide
7	ECT, Rhonda, John, David, Joe, Smith, Michae,l Mike
8	power
9	mvc, jpg, attach, meet, power, energy, Canada
10	calendard, standard, Monica, vacation, migration

Table 5.5: White textual clusters, our results compared to [12] results

Cluster	Ten most relevant words
1	schedules, hourahead, date 02, 02, date, dbcaps97data, hour, 02 hourahead hour, date 02 hourahead, start date 02
2	dbcaps97data, database, database alias dbcaps97data, unknown database alias, alias dbcaps97data unknown, dbcaps97data unknown database, cannot perform operation, dbcaps97data cannot perform, error dbcaps97data cannot, operation closed database
3	download failed manual, failed manual intervention, hour hourahead schedule, hourahead hour hourahead, hourahead schedule download, manual intervention required, schedule download failed, intervention, required, download
4	load, type, trans, id, load schedule, schedule, mkt type, trans type final, id enrj, mkt type trans
5	type, energy import export, import export schedule, id enrj ciso, general sql error, import export, 02 tie point, date 02 tie, engy type firm, final sc id
6	preferred, deal, assign deal number, cannot locate preferred, final individual interchange, individual interchange schedule, interchange schedule unable, locate preferred revised, matches final individual, preferred revised preferred
7	field, data, accept amount data, add try inserting, amount data attempted, attempted add try, data attempted add, error field accept, field accept amount, inserting pasting less
8	field, accept amount data, add try inserting, amount data attempted, attempted add try, data attempted add, error field accept, field accept amount, inserting pasting less, less data field
9	align, face verdana arial, verdana arial helvetica, face, align left, left, nbsp, align left face, left face verdana, energynewslive
10	outages, scheduled outages, scheduled, 713, 2002, sat 2002, sat, pager, thru, 853

Cluster	Most frequent and relevant words ([12] results)
1	Paso, iso, empow, ub, meet
2	schedule, detected, California, ISO, parsing
3	ub, employee, EPE, benefit, contact, ubsq
4	schedule, EPMI, NCPA, sell, buy, peak, energy
5	dbcaps97, data, failure, database
6	trade, pwr, impact, London
7	awarded, California, ISO, westdesk, Portland
8	error, pasting, admin, SQL, attempted
9	failure, failed, required, intervention, crawl
10	employee, price, ub, trade, energy

Table 5.6: Solberg textual clusters, our results compared to [12] results

Cluster	Ten most relevant words
1	inmarsat, telex, average, hoegh, master, bar, telex inmarsat telex, consumed, 158, fax
2	time, weekly report, weekly, houston time, dial numbers, passcode, report, 800 991 9019, 847 619 8039, domestic 800 991
3	power, dpc, mseb, project, dabhol, 2001, development, said, government, state
4	audrey, robertson, audrey robertson, 713, 646 2551 fax, 713 646 2551, 713 853 5849, audrey robertson 713, robertson 713 853, 5849 713 646
5	development, development development, gallons, cargo, paul, 2001, ect, days, barbo, cc
6	attached find weekly, week ending, report week ending, weekly report week, find weekly report, ending, 2001 saludos, weekly report, weekly, saludos
7	tk, 584, 874, inmarsat telex 584, tk tk, miles, nil, consumed nil, master mail master, hansen
8	questions, 345, 713 345, please, 345 5855 best, 5855 best regards, 713 345 5855, call 713 345, dpc project printed, free call 713
9	karolyn criado, thank karolyn criado, regarding last, karolyn criado 9441, questions regarding, questions regarding last, last weeks prices, regarding last weeks, prices thank karolyn, last weeks
10	thru, sat 2001, sat, outages, scheduled outages, ct, 2001, 2001 pt, 2001 london, 2001 ct

Cluster	Most frequent and relevant words ([12] results)
1	report, status, week, mmbtu, price, lng, lpg, capacity
2	tomdd, attach, ship, ect, master, document
3	London, power, report, impact, gas, rate, market, contact
4	dpc, transwestern, pipeline, plan
5	inmarsat, galleon, eta, telex, master, bar, fax, sea, wind
6	rate, lng, price, agreement, contract, meet
7	report, Houston, Dubai, dial, domest, lng, passcode
8	power, Dabhol, India, dpc, mseb, govern, Maharashtra
9	cargo, winter, gallon, price, eco, gas
10	arctic, cargo, methan

Table 5.8: Ybarbo textual clusters, our results compared to [12] results

Cluster	Ten most relevant words
1	2001, message, subject, sent, original, original message, steffes, james, steffes james, october
2	task, priority task due, task priority task, task start date, task assignment, start date, 2001 task start, due 2001 task, task due 2001, assignment
3	joc, news joc, cgi bin7 flo, joc cgi bin7, news joc cgi, cgi, news, joc online, online, mail
4	epsa, call, conference call, conference, affairs, ferc, nerc, rto, working group, regulatory affairs
5	recipient, intended, mail, attachments, mail including attachments, including attachments, intended recipient, ridertoe doc, epsa, doc
6	aps, ginger, paul, dernehl, kaufman, kaufman paul, james, 713, steffes james, october 2001
7	daily notice, daily notice 01, company, said, doc, mail including attachments, including attachments, notice, daily, dynegy
8	sce, jeff, ginger, cpuc, dernehl, dasovich, doc, mail including attachments, including attachments, ca
9	daily notice, daily notice 01, daily, notice, doc, 01, 01 epmi doc, notice 01 epmi, epmi doc, doc daily notice
10	mail including attachments, including attachments, ridertoe doc, aps, pge, including, doc ridertoe doc, ridertoe doc ridertoe, pge imbalance, imbalance

Cluster	Most frequent and relevant words ([12] results)
1	FERT, RTO, EPSA, NERC
2	market, FERC, Edison, contract, credit, order, RTO
3	FERC, report, approve, task, imag, attach
4	market, ee, meet, november, october
5	California, protect, attach, testimony, Washington
6	stock, billion, financial, market, trade, investor
7	market, credit, ee, energy, util
8	attach, gov, energy, sce
9	affair, meet, report, market
10	gov, meet, november, imbal, pge, usbr

Table 5.11: Steffes textual clusters, our results compared to [12] results

5.4 Forensic investigation of Enron scandal

In Section. 5.1 we gave a general background on the 'Enron case', and we pointed out some key characters that played an important role on the case. After seeing the general results and information we can induce using the framework, In this section we will analyze the dataset from an investigative prospect and search for meaningful information regard the 'Enron scandal'.

In this chapter we took the email archives of the two CEO of that period, whom have been highly implicated in the 'Enron scandal': Kenneth Lay and Jeffrey K. Skilling. A good strategy would be to analyze these archives from an investigative prospective, trying to discover interesting facts related to the Enron case and the related discoveries made in the past, which we already talked about (see Section. 5.1). So in the next 2 sections we will treat Jeffrey K. Skilling and Kenneth Lay individually, and we will list the most notable facts discovered using our framework.

5.4.1 Case study: Jeffrey K. Skilling

Two email accounts: There are two accounts associated to Jeffrey K. Skilling; jskilli@enron.com and jeff.skilling@enron.com. With respectively a contact strength equal to 270 and 4599 messages, which makes the second email account the most used one. The number of messages are relatively few considering the total quantity of messages, and the total number of messages sent are remarkably much less than the received one.

From the generated graph (see Figure. 5.8b), it looks like skilling was using each account for a different reason. The 'jeff.skilling@enron.com' is the one he used for his regular working duties with Enron members (nodes not surrounded by circles in the figure), while 'jskilli@enron.com' was a window for Enron external members to contact him, the nodes surrounded by a blue circle (e.g: aol.com, hotmail.com, sirius.com, swbanktx.com ...etc).

Family members: A very appealing node in the graph is 'markskilling@hotmail.com', so someone related to his family. Taking a look at 'markskilling@hotmail.com' we found out that he used to send emails from February 1999 to January 2000, and all the messages were directed to friends and family members ('Skilling' appeared in the accounts usernames). In some occasions he used to text friends and family members on their working account. Jeffrey Skilling has never sent any email to 'markskilling@hotmail.com' directly, instead he used other contacts as bridge (e.g: sherri.sera@enron.com). A very good notable email example from a "Skilling" family member to 'jskilli@enron.com' was received in '2000/12/13' with subject: 'CONGRATULATIONS!', the intent was congratulating Jeff Skilling on becoming the new Enron CEO.

Relations with other suspects: Looking at the relations with the other key characters of 'Enron scandal' we found out 3 different emails with Andrew Festow. A very interesting one is "FW: MD PRC Committee", an email that Fastow sent to Skilling talking about the importance of working with Ben Glisan, another important key character.

California energy crises If we take a look at the graphic representation for terms over time, we notice that "california" stands as one term on the top of the table in May 2001, selecting this word pointed out some interesting emails: an email by Skilling to all the Enron stuff in date:13/3/2001, explaining the "California energy crisis", and an email in 13/6/2001 to Skilling, Lay and Fastow reporting the crisis and the fact that they have done nothing to fix it.

5.4.2 Case study: Kenneth Lay

Internal (enron) and external users complaining accounts with 'enron.com' domain (internal) and with 'hotmail.com' (private), were sending their gripes and disappointments to Lay's account, after the 'enron scandal' arise. An extreme example is, a sarcastic email from one member of

the US Energy Services Inc ('usenergyservices.com' domain), who sent an email to Lay, after the scandal news, hoping that all the Enron directors (including Skilling and Fastow) spend a lot of time in jail.

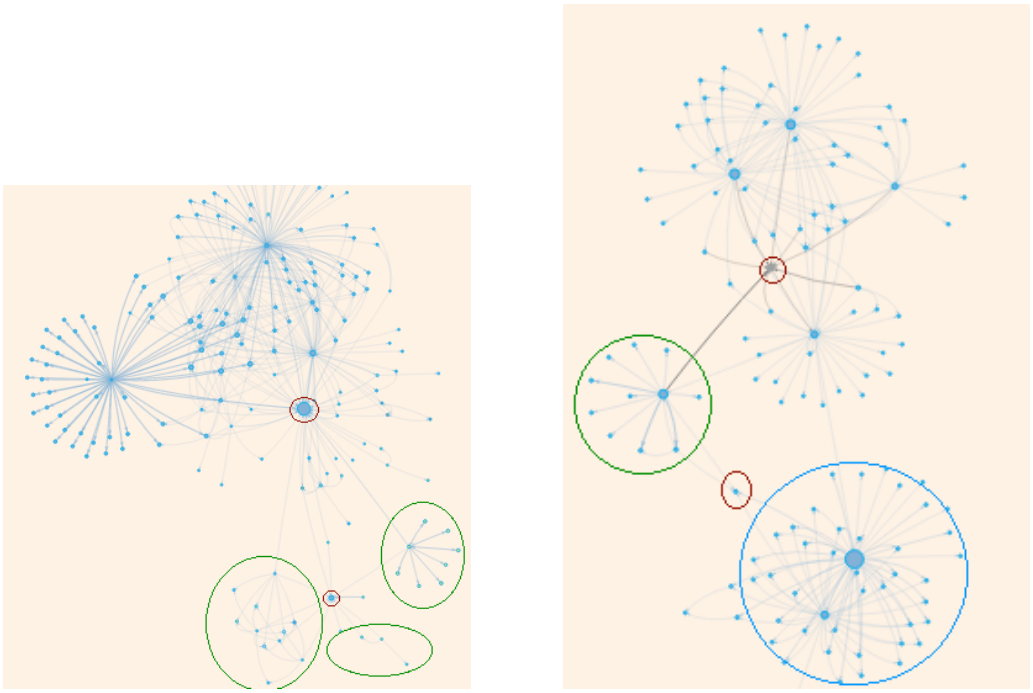
Two email accounts As for Skilling also in this case Lay have two different accounts: he used one for the internal Enron communications, and the second one for external contacts, with private (eg: 'yahoo.com' and 'hotmail.com') and business people (e.g: 'aol.com'), see Figure. 5.8a.

Lay succession plan Lay sent an email to a lot of Enron users, explaining his suggestion for Skilling as the next Enron CEO, saying that there will be no critical changes in the company management strategies.

Skilling leaving Enron Lay sent an email to microsoft member explaining the fact that Skilling left the position of Enron CEO for personal reasons, although he regrets his decision. Lay was assuring the fact that this departure will not have any effect with the relations between Enron and Microsoft.

Fastow not mentioned we don't have any relevant message talking about Andrew Festow.

Bankrupt and employees severance When looking at the graphic representation for terms over time, we notice that on the final months in the archive of Lay (Nov 2001, Dec 2001, and Jan 2002) the most relevant words became "bankruptcy", "employees", "consumers" ... etc. A further analysis of these terms reveled several emails addressing mr.Lay and asking him to correctly manage the money earned from selling Enron, and to keep in mind the rights of his employees.



(a) (b)

Figure 5.8: (a): The message traffic graph network of Lay, the circles surround: red for Lay accounts, green for all the non-Enron contacts. (b): The message traffic graph network of Skilling, the circles surround: red for Skilling accounts, blue for Enron external contacts, green for Skilling family contacts

Chapter 6

Conclusions

The main propose of this work was to create a final usable tool which can assist users in the analysis of email collections using efficient automated methods and data analysis techniques. This analysis can be accomplished by focusing on different behaviors, and through the mining of the data from different perspectives. The architecture of our tool was therefore perceived to be flexible and expandable for further analysis integrations and improvements of the current features. We mainly focused our approach on two different yet fundamental aspects: social behaviors and the textual content of the emails body.

In order to achieve successful results and a system with the desired characteristics, we studied several fields of interest, such like: text mining, forensic analysis, data elaboration, and data visualization techniques. We evaluated the most relevant features deployed by current real frameworks, and took note of the points we need to integrate in our framework, along with the introduction of new features which might benefit from the adoption of innovative techniques briefly used in currently diffused frameworks.

Social network analysis was the first and main field of interest, the main intent was elaborating the messages metadata and build two network graph representations: a relationship network that emphasizes the collaboration between different contacts as distinct clusters, and a message traffic network

to visualize the messages flow distribution in sending and receiving activities. The second analysis took in consideration the textual content, here we concerned our approach on applying text categorization methods. We used LSA (Latent Semantic Analysis) as a method of topics detection, and TFIDF text weighting scheme, to build a graphical representation of text relevance over the time.

To visually represent the analysis made we used network graphs and other 2D graphical schemes. This will facilitate the user interaction with the framework, in addition users will have the possibility to actively interact with the visualizations and the elaborations by the application of different filters.

We used the Enron email collections to test our framework functionality and usage, this process was conducted in two steps: firstly we evaluated the main features of our framework by taking as input random archives and demonstrating the potentiality of the framework. On the second phase we addressed the actual 'Enron scandal' case and focused our analysis on the real key characters effectively involved, trying to find correlations between the real juridical reports and the results obtained from the framework.

Bibliography

[1] D3 js, a javascript library for manipulating documents based on data. https://d3js.org/.

[2] vis js, a dynamic, browser based visualization library. http://visjs.org/.

[3] Frederic Baguelin, Christophe Malinge Solal Jacob, and Jeremy Mounier. Digital Forensics Framework - ArxSys dff (digital forensics framework), 2013. http://www.arxsys.fr/discover/.

[4] Alexei Barrionuevo. 10 enron players: Where they landed after the fall, 2006. http://www.nytimes.com/2006/01/29/business/businessspecial3/10-enron-players-where-they-landed-after-the-fall.html.

[5] M Basavaraju and R Prabhakar. A novel method of spam mail detection using text based clustering approach. *International Journal of Computer Applications*, 5(4):15–25, 2010.

[6] Michael Baur, Ulrik Brandes, Jürgen Lerner, and Dorothea Wagner. Group-level analysis and visualization of social networks. In *Algorithmics of large and complex networks*, pages 330–358. Springer, 2009.

[7] Benjamin Bengfort and Konstantinos Xirogiannopoulos. Visual discovery of communication patterns in email networks. 2015.

[8] Jay T Buckingham, Geoffrey J Hulten, Joshua T Goodman, and Robert L Rounthwaite. Using message features and sender identity for email spam filtering, March 1 2011. US Patent 7,899,866.

[9] Koutras Nikolaos Charalambous Elisavet, Bratskas Romaios. Email forensic tools: A roadmap to email header analysis through cybercrime use case. *Journal of Polish Safety and Reliability Association*, 7(1):21–28, 2016.

[10] Paraben Corporation. Paraben a universal platform for digital evidence. `https://www.paraben.com/`.

[11] Kristof Coussement and Dirk Van den Poel. Improving customer complaint management by automatic email classification using linguistic style features as predictors. *Decision Support Systems*, 44(4):870–882, 2008.

[12] Sergio Decherchi, Simone Tacconi, Judith Redi, Alessio Leoncini, Fabio Sangiacomo, and Rodolfo Zunino. Text clustering for digital forensics analysis. In *Computational Intelligence in Security for Information Systems*, pages 29–36. Springer, 2009.

[13] Vamshee Krishna Devendran, Hossain Shahriar, and Victor Clincy. A comparative study of email forensic tools. *Journal of Information Security*, 6(2):111, 2015.

[14] Nicholas Evangelopoulos, Xiaoni Zhang, and Victor R Prybutok. Latent semantic analysis: five methodological recommendations. *European Journal of Information Systems*, 21(1):70–86, 2012.

[15] Xiaoyan Fu, Seok-Hee Hong, Nikola S Nikolov, Xiaobin Shen, Yingxin Wu, and Kai Xuk. Visualization and analysis of email networks. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, pages 1–8. IEEE, 2007.

[16] Simson L Garfinkel. Digital forensics research: The next 10 years. *digital investigation*, 7:S64–S73, 2010.

[17] David Gefen and Kai R Larsen. Controlling for lexical closeness in survey research: A demonstration on the technology acceptance model.

[18] Andrea de Franceschi Gianluca Costa. Xplico open source network forensic analysis tool (nfat), 2013. `http://www.xplico.org/`.

[19] Rachid Hadjidj, Mourad Debbabi, Hakim Lounis, Farkhund Iqbal, Adam Szporer, and Djamel Benredjem. Towards an integrated e-mail forensic analysis framework. *digital investigation*, 5(3):124–137, 2009.

[20] SysTools Inc. Mailxaminer specialized email forensic tool, 2013. `https://www.mailxaminer.com/`.

[21] Deepak Jagdish. *IMMERSION: a platform for visualization and temporal analysis of email data*. PhD thesis, Massachusetts Institute of Technology, 2014.

[22] Kostiantyn Kucher and Andreas Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *Visualization Symposium (PacificVis), 2015 IEEE Pacific*, pages 117–121. IEEE, 2015.

[23] MIT Media Lab. Immersion a people-centric view of your email life, 2013. `https://immersion.media.mit.edu`.

[24] Robert Laurini. Geographic ontologies, gazetteers and multilingualism. *Future Internet*, 7(1):1–23, 2015.

[25] Vound Inc. LLC. Intella forensic search, ediscovery, and information governance. `https://www.vound-software.com/`.

[26] Fookes Software Ltd. Aid4Mail the accurate, fast way to migrate, archive and analyze email data. `http://www.aid4mail.com/`.

[27] Andri Mirzal. Clustering and latent semantic indexing aspects of the singular value decomposition. *arXiv preprint arXiv:1011.4104*, 2010.

[28] Kasula Chaithanya Pramodh and P Vijayapal Reddy. A novel approach for document clustering using concept extraction. 2014.

[29] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[30] Mithileysh Sathiyanarayanan and Nikolay Burlutskiy. Visualizing social networks using a treemap overlaid with a graph. *Procedia Computer Science*, 58:113–120, 2015.

[31] Rushdi Shams and Robert E Mercer. Classifying spam emails using text and readability features. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 657–666. IEEE, 2013.

[32] Michael Spranger and Dirk Labudde. Semantic tools for forensics: Approaches in forensic text analysis. In *Proc. 3rd. International Conference on Advances in Information Management and Mining (IMMM), IARIA. ThinkMind Library*, pages 97–100, 2013.

[33] Guanting Tang, Jian Pei, and Wo-Shun Luk. Email mining: tasks, common techniques, and tools. *Knowledge and Information Systems*, 41(1):1–31, 2014.

[34] Fernanda B Viégas, Scott Golder, and Judith Donath. Visualizing email content: portraying relationships from conversational histories. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 979–988. ACM, 2006.

[35] Visualware. EmailTrackerPro email tracer and spam filter. `http://www.emailtrackerpro.com/`.

[36] Raphael Volz, Joachim Kleb, and Wolfgang Mueller. Towards ontology-based disambiguation of geographical identifiers. In *I3*, 2007.

[37] Lidong Wang, Guanghui Wang, and Cheryl Ann Alexander. Big data and visualization: methods, challenges and technology progress. *Digital Technologies*, 1(1):33–38, 2015.

[38] Chun Wei, Alan Sprague, Gary Warner, and Anthony Skjellum. Mining spam email to identify common origins for forensic application. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1433–1437. ACM, 2008.

[39] CMU William W. Cohen, MLD. Enron email dataset. `https://www.cs.cmu.edu/~./enron/`.

[40] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.