Deep Email Miner Application

Software Manual

Johannes Mager

University of Technology, Sydney

sf@johannes-mager.de

June 23, 2006

Contents

1 Introduction

The *Deep Email Miner Application* is a software program for the multistaged analysis of an email corpus. Social network analysis and text mining techniques are connected to enable a new in depth view into the underlying information.

It allows the visualization of an email corpus as a social network, where the nodes are email senders/recipients and the edges represent the email traffic. The network can be filtered and transformed, clusters can be detected and several network measures and rankings applied. The email traffic of each node and edge can be displayed, filtered and automatically grouped to email treads. Emails can be labelled and analysed using the built-in word vector tool.

All data is loaded from a database, with the sample data set being the freely available email corpus of the Enron company. The software is based on the Java programming language, uses several open software libraries and is distributed as open source.

In this User Manual, the version *v1* of the *Deep Email Miner Application* and its usage will be thoroughly described. After the program's features have been introduced, the third section describes a sample analysis procedure for a new email corpus. Section four provides a closer look into the program's components. In the appendix, the database scheme and the contents of the optional properties file will be defined.

Please note, that the current version of the *Deep Email Miner Application* does not yet contain any anonymisation features for respecting the privacy of the users. These features will be introduced in future versions. For a discussion of this issue, please refer to: Peter Fule and John F. Roddick, *Detecting privacy and ethical sensitivity in data mining results*, 2004

2 Features

2.1 Social Network Analysis

- Create a social network of an email corpus with the employees as vertices
- Edges indicate email traffic between two employees
- Find clusters of employees
- Rank nodes/edges on different centrality measures
- Display network statistics

2.2 Email Analysis

- Detect email threads
- Display emails
- Add and remove network filters based on different email attributes and the number of emails sent
- Display email corpus statistics

2.3 Text Mining

- Label emails as business/private
- Create a word list from the labelled emails
- Create the word vectors of the labelled emails

2.4 Input/Output

- Load the network and single emails from a database
- Cache email bodies to balance the amount of database-interaction and the program's memory usage
- Load/Save email labels to file
- Export word vectors to .csv format
- Export network to Pajek .net format

2.5 Software Features

- Efficient enough to handle email corpora with several hundred people
- Written in Java, runnable everywhere
- Additional version as Win32 executable available
- Configurable with a properties-file

3 Analysing an Email Corpus

In this section, a sample analysis procedure of an email corpus will be explained. A more detailed description of all functions used is available in the section Program Components (4).

3.1 Download and Installation

The runnable *Deep Email Miner Application* binaries are available for download from the Deep Email Miner homepage as a Windows `.exe` or as a Java `.jar` archive. Both versions require the Java Runtime Environment 5.0 or newer installed on the computer (see Java homepage).

In most cases, the application can be loaded by double-clicking the program file. To run it from the command line, simply use the following command:
`java -jar DeepEmailMiner-xx.jar`.

Please note: For the `.jar`-version and complex email networks, it may be necessary to increase the maximum available memory size. Use the following command: `java -jar -Xmx200M DeepEmailMiner-xx.jar`.

Tip: Many parameters of the *Deep Email Miner Application* can be adjusted by using a configuration file as explained in the appendix (C).

3.2 Loading a Network

On program start, no network is loaded. For a quick insight in the *Deep Email Miner Application*'s functionality, a sample graph can be loaded from the File-Menu.

If a email database has been configured for the program (see appendix (B)), the email corpus can be loaded from this database via the File-Menu.

3.3 Analysing the Social Network

Once the email network is loaded, different techniques are available to gain an overview over the data and for the deeper analysis of this corpus.

To change the view on the network, it can be zoomed, rotated and sheared using the mouse. The unconnected vertices and the vertex labels can be switched on and off and single vertices can be moved around.

An overview over all vertices and edges of the network can be displayed using the list windows. Clicking an entry with the mouse highlights it's position in the network. A special kind of these lists are the rankings. In addition to displaying the network elements, they also compare and rank them using one of several metrics.

If the opened network is too complex, or certain emails shall be excluded (eg. exclude emails after a certain date), the network can be filtered using the filtering tab. If an interesting network state is found, it can also be exported to the *Pajek .net* format and further analysed in other graph programs.

To find groups of strongly connected vertices (*clusters*), enable the clustering mode and drag the slider further to the right. Edges of the network are removed successively and each time a new cluster forms, the color of it's vertices changes. To have a better overview, cluster grouping can be enabled. The cluster list lists all current clusters and is updated permanently. By clicking on one of the entries, the entire cluster is highlighted in the network.

3.4 Email Analysis

The emails contained in an edge and the internal / external incoming / outgoing (Glossary A) emails of each vertex can be viewed by right-clicking on a single edge or vertex. Double-clicking on the email subject shows the actual email text. To view all recipients of the email, simply use the *All*-Button.

The thread list groups the internal emails to threads in the original order. Double-clicking on one of the threads shows it's emails. Note: if the data was very sparse (as in the sample Enron email corpus), most threads will be incomplete.

3.5 Text Mining

For the future integration of machine learning into the *Deep Email Miner Application*, the emails already can be labelled into two categories: *business* and *private*. By labelling email threads, more than one email is labelled at once. To make sure that the label data will not be lost, you have to create a new output file first. All new labels will automatically be saved to this file. The labelled emails can be reviewed via the text mining menu. Note: A sample file of labelled Enron emails is available from the Deep Email Miner homepage.

The integrated statistical text analysis component can decompose the email texts and, using the labels, create the word list and the word vectors of all labelled emails. Note: The single words of the emails are processed by a stemming algorithm and therefore transformed to a base form.

The word vectors can also be exported to the `.csv`-format and loaded using Excel or other programs.

4 Program Components

The following section provides a detailed description of all of the *Deep Email Miner Application*'s components and how to use them.

4.1 Network Window

4.1.1 Elements

The network window is the main part of the application, the place where the actual email network is displayed. The network's components are defined as follows:

- Each vertex symbolises an employee (Glossary A) and is by default labelled with the employee's first name (can be switched on/off via the network tab (4.2.1)). If the mouse cursor is above a vertex, the employee's full name and email address are displayed in addition. The different vertex colors show whether the vertex is unconnected (Glossary A) (grey) or currently picked (yellow).
  After a right click of the mouse on a vertex, a context menu appears, allowing the user to view all internal / external incoming / outgoing (Glossary A) emails of the employee.
- A directed edge from *employee 1* to *employee 2* symbolises that *employee 1* sent one or more emails to *employee 2*, either as TO-, CC- or BCC-recipient (Glossary A). The opacity of an edge's color indicates the number of emails sent. If an edge is picked, it is colored red.
  After a right click of the mouse on an edge, the context menu allows the user to view all emails contained in this edge.

An overview over all vertex/edges in the current network can be obtained using the Vertex List (4.5.1) and Edge List Window (4.5.2).

4.1.2 Transforming the View

Using the mouse, the view on the displayed network can be modified in many different ways.
Please note: All these transformations to the current view require that the *Mouse Mode* is set to *Transforming* in the Network Tab (4.2.1).

- Dragging the network with the mouse changes it's position in the network window.

- Dragging the network while pressing the shift-button on the keyboard rotates the network.
- Dragging the network while pressing the control-button shears the network.
- Scrolling with the mouse wheel zooms the network in and out. (This can also be achieved with the zoom buttons in the Network Tab (4.2.1)).

4.1.3 Moving Vertices

If the *Mouse Mode* is set to *Picking* in the Network Tab (4.2.1), single or multiple vertices can be selected and moved around.

- Clicking on a vertex selects this vertex and deselects all others.
- Clicking on an empty spot in the network window removes all selections.
- Clicking on a vertex while holding the shift-button adds/removes this vertex from the current selection.
- Clicking on a vertex while holding the control-button centers the view of the network on this vertex.

4.1.4 Info Area

The *info area* in the left bottom corner of the network window provides instant access to some important network/program parameters like the number of filtered vertices, the current clustering and labelling states, the application's current memory usage etc.

4.2 Function Tabs

The three *function tabs* at the bottom of the network window allow the configuration of the most important network parameters as well as the clustering and filtering settings.

4.2.1 Network

The *network tab* provides access to the most important parameters defining the view of an email corpus:

- The mouse mode can be set to Transforming (4.1.2) or Picking (4.1.3) as explained above.

- Unconnected vertices (Glossary A) can be shown or hidden.
- The vertex labels can be switched on and off.
- The network layout can be recalculated.
- The network can be zoomed in and out.

4.2.2 Clustering

The *clustering mode* allows the user to find subgroups of employees that are closer connected among each other than to employees from outside this group. In order to find such groups, edges are consecutively removed from the network. The selection of these removed edges is based on an important social network measure, the Betweenness Centrality (Glossary A).

After enabling the clustering mode, the user can remove edges using the slider control. If the slider is moved to the right side, all edges are removed from the network. While removed edges are drawn thinner, the vertices are continually re-colored to indicate their membership in a specific cluster. Unconnected vertices are painted grey. If the network is filtered, the clustering mode is automatically disabled.

To improve the view on the clustered network, the different clusters can be grouped together. An overview over all current clusters can be obtained using the Cluster List Window (4.5.3).

4.2.3 Filtering

As email corpora can be based on the data of several hundred employees (Glossary A), the resulting network is very likely to contain several thousand edges. Displaying all this data at once results in a very complex network, making an analysis very difficult.

The filtering mechanism allows the user to select criteria about whether or not emails are included in the corresponding edges of the network. In addition, the number of emails an edge must contain to be displayed can be altered using the slider control.

The filtering attributes are:

- The recipient type (Glossary A) with which this email was sent from the sender to the recipients. Only the selected recipient types are included.
- The date on which the email was sent.

On pressing the *Apply*-button, the graph is filtered.

4.3 Menu Bar

4.3.1 File

In the file menu, the sample graph (4.7) can be loaded. *Open from database* opens an input window requesting the connection data for the database access. Default-values for the database server name and the database name can be set in the configuration-file (C). The application will now load the email corpus from the database and display it's social network. (For more information, see Configuring the database (B)).

Using the option *Save network*, the network can be exported to the *Pajek .net* format of the widely used Pajek network analysis software (For more information, see the Pajek homepage).

4.3.2 Rankings

See section Ranking (4.4).

4.3.3 Lists

See section List Windows (4.5).

4.3.4 Statistics

In the statistics menu, statistics about the network and it's components can be displayed for either the filtered graph as currently displayed in the network window, or the whole graph, ignoring any filter settings.

4.3.5 Textmining

See section Text Mining (4.9).

4.3.6 Windows

In the windows menu, all currently opened program windows are listed and can be displayed by click or keyboard shortcut.

4.3.7 Help

In the help menu, the about window with information about the program and it's authors can be displayed.

4.4 Ranking

Rankings of edges/vertices assign a distinct value to each edge/vertex and order the elements by this value. The results are displayed in a small list. Additionally, the graph elements are labelled with their rank.

4.4.1 Vertex

Vertices can be ranked on their number of edges, their number of internal (Glossary A) emails and their Betweenness Centrality (Glossary A).

4.4.2 Edge

Edges can be ranked on their number of internal (Glossary A) emails and their Betweenness Centrality (Glossary A).

4.5 List Windows

The *List Windows* provide an overview over the email network's components. The lists can be sorted ascending/descending on each of their columns by clicking and control-clicking on the column titles.

4.5.1 Vertex List

The *Vertex List* lists all vertices of the current network, representing the employees (Glossary A). If the network is filtered, this list is updated automatically.

Selecting one or multiple table entries with the mouse selects the corresponding vertices in the network window as well (to add/remove further entries, hold control on the keyboard).

4.5.2 Edge List

The *Edge List* lists all edges of the current network together with their current number of total and filtered emails. If the network is filtered, this list is updated automatically.

Selecting one or multiple table entries with the mouse selects the corresponding edges and their start/end vertices in the network window as well (to add/remove further entries, hold control on the keyboard).

4.5.3 Cluster List

If the Clustering Mode (4.2.2) is enabled, this window lists all current clusters and continually updates the view every time the clusters change. In this list, unconnected vertices are grouped together.

Selecting one or multiple table entries with the mouse selects the corresponding vertices in the network window as well (to add/remove further entries, hold control on the keyboard).

If the network is filtered or a new network is loaded, this window closes automatically.

4.5.4 Thread List

The *Thread list* presents the results of the *Deep Email Miner Application*'s integrated email thread detection. Based on several criteria, emails are added to different threads, summarized under their common subject.

Double-clicking on one of the subjects opens an email list window (4.6.1) where all of the thread's emails are listed and can be displayed.

4.6 Email Windows

4.6.1 Email List

Email List Windows can be displayed by right-clicking edges/vertices a network-component with the mouse or by double-clicking on entries in the email thread list.

Email lists can be sorted by clicking and control-clicking on the column titles.

Double-clicking on one of the emails opens an email viewer (4.6.2) with this email.

4.6.2 Email Viewer

The *Email Viewer Window* displays a single email with all it's data. While the recipients may be hidden due to performance issues, they can be displayed easily via the *Display-/All*-button.

4.7 Sample Network

The *Deep Email Miner Application*'s sample network is a small network of 9 vertices, 15 edges and a total of 25 emails. Still, it is fully suitable for getting a quick impression of many of the *Deep Email Miner Application*'s functions.

4.8 Labelling

The email labelling-mechanism allows the user to classify internal emails in one of the two categories *private* and *business*. In future versions, this mechanism will be generalized to user-specified categories and the labels will be used for training a machine learning component.

A priority of the labelling mechanism is that the labelled emails will not be lost in case of a termination of the program. Therefore, the user has to choose an empty output file first in which the labels will be saved. Another possibility is to load and extend an existing labelling file. After labelling a certain number of emails and on closing the labelling window, the labels will be automatically saved to this file.

Two labelling mechanisms are available: *Thread labelling* presents the detected threads one after another and therefore allows the user to label more than one single email at once. If all threads have been labelled, the labeller automatically switches to the other mechanism, the *email labelling*: randomly chosen internal emails are presented.

4.9 Text Mining

The text mining functionality of the *Deep Email Miner Application* v1 is very limited. When email labels are loaded, a word list can be displayed, listing the number of occurences of (stemmed) words in all labelled emails and their categories. The word vectors of the labelled emails can be calculated, displayed and exported to a file in `.csv`-format for further processing.

A Glossary

Betweenness Centrality is a measure used in social network analysis: Edges that appear on many shortest paths between other vertices have a higher betweenness centrality.

Employees in the context of this program are all people listed in the database's (B) `employeelist` table.

External emails are sent from or received by a person not listed as an employee.

Incoming emails of an employee have the employee's email address in at least one of the recipient fields.

Internal emails are sent from one employee to another.

Outgoing emails of an employee are all emails sent by him.

Recipient Fields are the *TO*, *CC* and *BCC* fields of an email

Unconnected is a vertex without incoming and outgoing edges (also called *isolated*).

B Configuring the database

The *Deep Email Miner Application* v1 supports the open source database system MySQL.

The reference database for the *Deep Email Miner Application* is the preprocessed Enron email corpus by Jitesh Shetty and Jafar Adibi from the University of Southern California (Download is available on their homepage).

For an email database to be used by the *Deep Email Miner Application*, the following tables have to exist:

An `employeelist` table with the first name (`firstName`), last name (`lastName`) and email address (`Email_id`) of the employees.
The entries of this table define the vertices of the email network.

```
CREATE TABLE employeelist (
    firstName varchar(31) NOT NULL,
    lastName varchar(31) NOT NULL,
    Email_id varchar(31) NOT NULL,
    UNIQUE KEY Email_id (Email_id));
```

A `message` table with an unique ID (`mid`), the sender email address (`sender`), the date (`date`), the email subject (`subject`) and the email body (`body`) of each email.

```
CREATE TABLE message (
    mid int(10) NOT NULL,
    sender varchar(127) NOT NULL,
    date datetime,
    subject text,
    body text,
    PRIMARY KEY (mid));
```

A `recipientinfo` table with an unique ID (`rid`), the message ID of the corresponding email (`mid`), the recipient type (`rtype`) and the email address of the recipient (`rvalue`) of each recipient of each email.

```
CREATE TABLE recipientinfo (
    rid int(10) NOT NULL,
    mid int(10) unsigned NOT NULL,
    rtype enum('TO','CC','BCC'),
    rvalue varchar(127),
    PRIMARY KEY (rid));
```

As the queries for building the initial email network from the database are very complex, it is highly recommended to create the following database indices:

- `employeelist.Email_id`
- `message.sender`
- `recipientinfo.mid`
- `recipientinfo.rvalue`

C Using a configuration file

Using a configuration file, many parameters of the *Deep Email Miner Application* can be adjusted according to the user's preferences.

To be detected and used on program start, a configuration file with the name `config.properties` has to lie in the same directory as the program (For users starting the program from the command line, the file has to lie in the shell's current directory when the program is started). If the file was loaded, `Properties file found` is displayed in the Info Area (4.1.4).

The file (a sample can be downloaded from the Deep Email Miner homepage) has the following format: Lines beginning with a `#` are comments and ignored, all other non-empty lines should have the form `key=value`, where `key` is one of the program's parameter names.

C.1 General Settings

`startGraph` The graph loaded upon program start (`sample/empty`)

`minimumEmailsInThread` The minimum number of emails to form a thread (default 2)

`maxDaysBetweenThreadMessages` The maximum number of days between two successive messages in one thread (default 30)

`emailCacheSize` The maximum size of the email cache (default 5000)

C.2 Filter Settings

`initialEmailFilter` The initial email filter: show edges with more than `x` emails (default 50)

`filterStartYear` The start year displayed in the email filter (default 1996)

`filterEndYear` The end year displayed in the email filter (default current Year)

`edgeNumberWarning` If the graph gets bigger than `x` edges, show warning (default unlimited)

C.3 Labelling Settings

`emailLabelPercentage` Maximum percentage of labelled internal emails (default 10)

`autoSaveInterval` Save email labels each `x` processed messages (default 50)

C.4 Database Settings

The connection data default values are the settings of the database driver

`db.defaultServerName`

`db.defaultServerPort`

`db.defaultDatabaseName`

C.5 Textmining Settings

`wordListMinimumOccurence` The minimum number of occurrences a word must have to be included in the textmining word list (default 5)

`wordListMaximumOccurence` The maximum number of occurrences a word can have to be included in the textmining word list (default unlimited)

C.6 Debug Settings

Please note: These parameters have been created for the debug process and may decrease the analysis capacity of the *Deep Email Miner Application*.

`debug.employeeNumber` Limit the number of processed employees during database-loading (default unlimited)

D Acknowledgements

Many thanks go to Simeon and Debbie from the E-Markets Group of the University of Technology, Sydney (see homepage) for all the help, tips and advices during the development of the *Deep Email Miner Application*.

The *Deep Email Miner Application* uses a variety of open source Java projects, where the most important are:

- The amazing *Java Universal Network/Graph Framework JUNG* (see homepage) from the University of California, Irvine, used for most of the graph drawing and visualisation functionality.
- The *WVTool* (see homepage) from the University of Dortmund, Germany, for the statistical language modelling functionality.