An Empirical Study on the Risks of Using Off-the-Shelf Techniques for Processing Mailing List Data

Nicolas Bettenburg Emad Shihab Ahmed E. Hassan
Software Analysis and Intelligence Lab
Queen's University
Kingston, Canada
{*nicbet, emads, ahmed*}*@cs.queensu.ca*

Abstract

Mailing list repositories contain valuable information about the history of a project. Research is starting to mine this information to support developers and maintainers of long-lived software projects. However, such information exists as unstructured data that needs special processing before it can be studied. In this paper, we identify several challenges that arise when using off-the-shelf techniques for processing mailing list data. Our study highlights the importance of proper processing of mailing list data to ensure accurate research results.

1. Introduction

Electronic mail is an established form of communication in networked computing environments. Mailing list software distributes messages to a predefined list of recipients and is widely used in software development. There it aids day-to-day development and enables communication between project stakeholders, e.g., developers and users. Messages sent over these mailing lists contain a multitude of information on the project, such as important development decisions, discussions of the source code, and support requests. Software maintainers can use this information to study corrective activities [17], developer communication [12], or knowledge recovery [16].

Although mailing list data is often readily available online, transforming the data into a structured format that is suitable for subsequent analysis is a challenging task. Messages are often stored in email archives and need to be extracted before they can be used. However, mailing list archives contain duplicate and invalid data, stored in raw formats, which need further processing. Additionally, up to 98.4% of electronic messages contain noise that threatens the applicability of text mining approaches [15]. Researchers need to be aware of potential pitfalls and take special care before using the information mined from mailing list archives.

In this paper we identify difficulties that arise when processing mailing list data. These difficulties are present in most stages of the mining process, such as data collection, data extraction and information processing. Previous research has noted the presence of several challenges, but documented them only loosely, as they are a by-product of the research work conducted, rather than the main scope. Mining raw mailing list data yields potential risks to the accuracy of research results and should be avoided.

The rest of the paper is organized as follows. In Section 2 we highlight the risks of using unclean mailing data by an example mailing list analysis task. In Section 3 we present challenges that arise when using off-the-shelf techniques for mining of mailing list data. We present the work related to our study in Section 4 and conclude our work in Section 5.

2. Motivating Example

Summaries of recent discussions on the mailing list can be useful for decision makers to monitor the development progress and to identify topics of high interest, to recover knowledge about design decisions, an to aid the maintenance of legacy systems.

Although mailing list data is stored in a textual way, which humans can easily read and understand, using this data as-is in content-based analyses yields hidden, yet severe risks for the validity of the obtained results.

In this example we use tag clouds, a concept from information retrieval, to visualize the contents of a discussion thread. Tag clouds display the most frequent terms weighted by font size and color. The larger and more visible a term is presented in a tag cloud, the higher its semantic value for the text.

Figure 1 shows two tag clouds summarizing the contents of the same discussion thread on the PostgreSQL mailing list with the topic *"Explicit config patch 7.2B4"*, starting at December 16th, 2001. This discussion centers around the possibility of passing command line arguments to the PostgreSQL server executable, which allow the user to specify the locations of the server's configuration files, because many Linux distributions, besides Debian, scatter configuration files around in the file system.

The first cloud, presented in Figure 1a, is generated using the contents of the email messages that form the discussion thread as-is, i.e., without prior processing of the message bodies. The second cloud, presented in Figure 1b, is generated from the same email messages, however the messages were cleaned up significantly by removing attachments, signatures and quotations, as well as transforming all remaining parts into English language text.

Comparing both tag clouds, we can see that the tag cloud generated from uncleaned mailing list data contains a large amount of noise, which renders the interpretation of the discussion's contents a challenge. On the opposite, the summary produced from the cleaned discussion thread is much more helpful in giving a good idea of the contents of the discussion.

3. Processing Mailing List Data with Off-the-Shelf Techniques

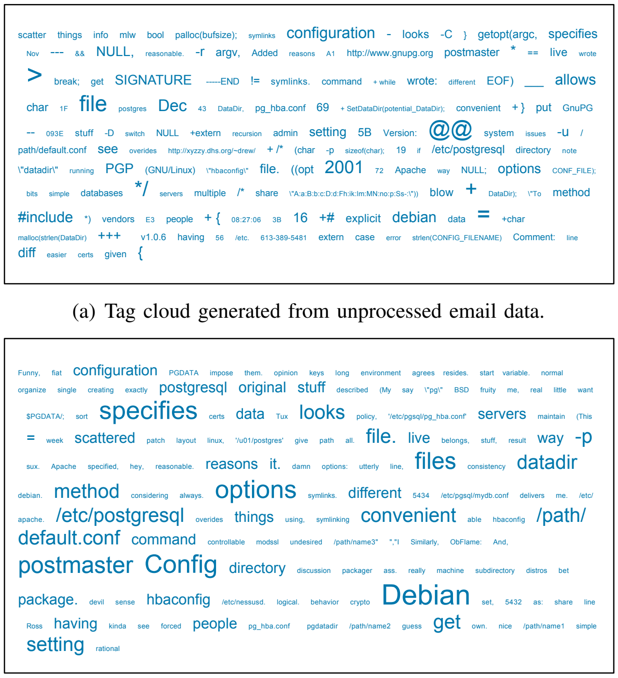
Name of Challenge	Level of Automation	Impact on Quality
Message Extraction	Automated	low
Duplicate Removal	Automated	high
Language Support	Automated	medium
MIME/Attachments	Automated	high
Quotes/Signatures	Semi-Automated	high
Thread Reconstruction	Semi-Automated	high
Resolving Identities	Semi-Automated	high

Table 1. Overview of challenges presented.

In this section we discuss challenges with using off-the-shelf techniques for mining mailing list data. An overview is presented in Table 1. For each challenge we assign a notion of automation and impact on data quality. Some challenges presented cannot be addressed in a completely automated manner and need manual tuning before reliable results can be obtained. We denote these as semi-automatable challenges. From a combination of automatability and the impact on the data quality we gain an intuition of the overall severity associated to each challenge.

3.1. Extracting Messages

Many open-source software projects store the messages of their mailing lists in *mbox* files [14], which represent textual databases that contain linear sequences of electronic messages. These messages need to be extracted before they can be analyzed. However, the extraction process requires knowledge about the structure of the archive. Additionally, competing MBOX specifications disagree on the format of the mail archive. Both the performance of extraction tools and the deficiencies of erroneous mailing list archives have an immediate impact on the quality and quantity of the extracted data.


scatter things info mlw bool palloc(bufsize); symlinks configuration - looks -C } getopt(argc, specifies
Nov --- && NULL, reasonable. -r argv, Added reasons A1 http://www.gnupg.org postmaster * == live wrote
> break; get SIGNATURE -----END != symlinks. command + while wrote: different EOF) ___ allows
char 1F file postgres Dec 43 DataDir, pg_hba.conf 69 + SetDataDir(potential_DataDir); convenient + } put GnuPG
-- 093E stuff -D switch NULL +extern recursion admin setting 5B Version: @@ system issues -u /
path/default.conf see overides http://xyzzy.dhs.org/~drew/ + /* (char -p sizeof(char); 19 if /etc/postgresql directory note
\"datadir\" running PGP (GNU/Linux) \"hbaconfig\" file. ((opt 2001 72 Apache way NULL; options CONF_FILE);
bits simple databases */ servers multiple /* share \"A:a:B:b:c:D:d:Fh:ik:lm:MN:no:p:Ss-:\")) blow + DataDir); \"To method
#include *) vendors E3 people + { 08:27:06 3B 16 +# explicit debian data = +char
malloc(strlen(DataDir) +++ v1.0.6 having 56 /etc. 613-389-5481 extern case error strlen(CONFIG_FILENAME) Comment: line
diff easier certs given {

(a) Tag cloud generated from unprocessed email data.

Funny, fiat configuration PGDATA impose them. opinion keys long environment agrees resides. start variable. normal
organize single creating exactly postgresql original stuff described (My say \"pg\" BSD fruity me, real little want
$PGDATA/; sort specifies certs data Tux looks policy, '/etc/pgsql/pg_hba.conf' servers maintain (This
= week scattered patch layout linux, '/u01/postgres' give path all. file. live belongs, stuff, result way -p
sux. Apache specified, hey, reasonable. reasons it. damn options: utterly line, files consistency datadir
debian. method considering always. options symlinks. different 5434 /etc/pgsql/mydb.conf delivers me. /etc/
apache. /etc/postgresql overides things using, symlinking convenient able hbaconfig /path/
default.conf command controllable modssl undesired /path/name3" ","I Similarly, ObFlame: And,
postmaster Config directory discussion packager ass. really machine subdirectory distros bet
package. devil sense hbaconfig /etc/nessusd. logical. behavior crypto Debian set, 5432 as: share line
Ross having kinda see forced people pg_hba.conf pgdatadir /path/name2 guess get own. nice /path/name1 simple
setting rational

(b) Tag cloud generated from processed email data.

Figure 1. Summarizing the contents of the same discussion thread using tag clouds generated from unclean and cleaned mailing list data.

3.2. Removing Duplicates

One essential part of any cleaning process in data mining involves the identification and removal of duplicate data [10]. This step is of utmost importance when the mined data is used in aggregation functions or frequency analyses. Duplicate entries will result in false or potentially misleading results. The 3 main sources of duplicate messages on mailing lists are:

1) *Network problems*, i.e., timeouts, can cause a message to be sent multiple times.
2) *Software errors* in the mailing list software can cause messages to be recorded multiple times.
3) *Accidental resubmission* (e.g., a user clicked a "send" button multiple times) can also result in duplicate messages to be transferred to the mailing list.

Solutions to this challenge, e.g., similarity measures like hashing or near-miss identification, can easily be automated.

3.3. Handling Multiple Languages

Since geographically distributed software development is increasing in both open-source [8] and industry [7], mailing lists are used for communication of a multitude of developers with different cultural backgrounds and languages. Character encodings specify how text in the writing systems of different languages is represented in a binary form [18]. Problems arise when the encoding of a message is ignored during the data mining process. For instance, the name "Réné" encoded in a French character set would be transformed to "Rn" when treated as English text. In order to safeguard the mined information from data loss, it is important to determine the appropriate encoding and safely translate text to an encoding like Unicode, which can handle multiple languages simultaneously.

Existing internationalization solutions like the MOZILLA character encoding detection algorithm can be used to robustly unify multi-language archives automatically.

3.4. Handling MIME Messages and Attachments

While plain text emails have the advantage of maximizing compatibility, MIME messages offer the ability to stylize the text and store extra information. Additionally to formatting, the MIME standard also specifies *attachments*. With attachments, users can transmit additional non-text data together with their messages, e.g., documents like spreadsheets, or electronic fingerprints.

While the MIME extension allows for specialized and custom styled messages that can contain binary data, it comes at the cost of making the data mining approach more challenging, as extra mechanisms are needed to handle these contents. A number of different composite MIME-types exist, that have implicit semantics for the message contents [4].

Stylized messages need to be translated from html or rich text to a plain text format, attachments need to be separated from the messages' contents and stored separately, and composite type messages need to be handled according to their implicit semantics.

3.5. Removing Quotes and Signatures

In mailing list discussions, users usually refer to text from a previous participant by quoting parts of the original message to give more context and meaning to their contributions, as S. Hambridge from Intel states in his network etiquette guidelines [6]:

> *"If you are sending a reply to a message or a posting be sure you summarize the original at the top of the message, or include just enough text of the original to give a context."*

However, the additional text might not be desirable for text- and data mining approaches, as it is redundant information that has already been encountered before. Quoted text typically begins with one or more ">" signs at the beginning of a line - one for each level of quotation - and is easily removed automatically.

Signatures are typically used as a "soft" proof of identity and to indicate that no more text is following in a message [6]. Signatures contain a variety of artifacts, such as contact information, text graphics, famous quotes and trivia. For instance, Google Mail [5] can be set up to include a random quote as a signature when sending email. Many free email services also add advertisements as a signature when a message is sent. As a result, the information in a signature block is often repetitive and unrelated to the message. Figure 2 illustrates a sample signature from a participant on the PostgreSQL developer's mailing list.

```
                                                     ________________________
.  .  _  ._  . .   .__    .  . ._. .__ .   . . .__  | Neighbourhood Coder
|\/| |_| |_| |/    |_     |\/|  |  |_  |   |/  |_   |
|  | | | | \ | \   |__ .  |  | .|. |__ |__ | \ |__  | Ottawa, Ontario, Canada
One ring to rule them all, one ring to find them, one ring to bring them all
and in the darkness bind them...
```

Figure 2. Sample signature extracted from a message on the PostgreSQL mailing list.

Current solutions to this challenge exist only as semi-automated tools or processes that need to resort to manual inspection and fine-tuning of parameters to yield good results.

3.6. Reconstructing Discussion Threads

An email *discussion thread* is a set of messages that are logically related, e.g., multiple answers to a question. The messages in the set form a tree-shaped hierarchy. Whenever a user participates in the discussion, his message becomes a child of the message he is replying to. The initial message starting a new topic is the root node.

However, mailing list archives commonly store messages based on their temporal order rather than their logical grouping. As such, the hierarchical order has to be reconstructed after the messages have been extracted.

As an additional challenge, a user's email client is responsible for storing unique identification information on messages in a special header field. This header is optional, so in practice one cannot rely on threading information to be present. For instance, the MICROSOFT Outlook [11] email client did not implement a message-id header until its latest version (Outlook 2007).

Current solutions to this challenge exist only as semi-automated processes that need to resort to manual inspection.

3.7. Resolving Multiple Identities

Some participants use multiple email addresses when taking part in discussions on a mailing list [1]. These addresses are aliases for individual personalities and should be resolved before using the data. Ignoring this problem can lead to problems when doing quantitative and social analyses.

We know of no fully-automated solution to this problem that can reliably detect and remove signatures from email messages.

4. Related Work

The challenges presented in this work have many implications for applications of mailing list data mining in research. In the past, many of these challenges have been described only anecdotical or as side-notes.

Bird et al. mine mailing lists to study social networks [1], [2]. They identify the multiple alias problem and propose the use of a clustering algorithm to merge identities.

Herraiz et al. identify that mining repositories of open-source projects is a challenging task and propose general approaches to mining these repositories [8], [13]. The ml-stats tool used for their studies on GNOME mailing lists, mines information from email headers.

Kolcz et al. use text-mining approaches to detect near-duplicate email messages for spam identification [9].

Carvalho et al. use machine learners to identify signatures and quotations in email messages [3]. While this method can achieve good results, it needs a manual training step and sufficiently clean training data to perform well.

Tang et al. propose methods for cleaning plain text email messages, in order to make them accessible for text-mining and information retrieval [15]. Their work focusses on text transformation for natural language processing.

5. Conclusions

Mailing lists contain valuable information for maintainers of long-lived software projects. In order to make this information accessible for subsequent analysis steps it needs to be processed first. Many mailing lists document multiple years of project development. However, the email technologies that produce this mailing list data have changed several times over the past decade. As such, mailing lists contain a conglomeration of messages from different revisions of the email format. Using off-the-shelf techniques to process this data naively yields many risks for the validity of the resulting information.

Yet, for many of the presented issues no perfect, automated solutions exist. Email messages are substantially different from the much cleaner text sources used in related research areas like information retrieval. As such many of the text cleaning techniques used in text-mining and information retrieval cannot be readily applied to email communication. Hence, we see an opportunity for future work to refine mailing list data processing techniques.

References

[1] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan, "Mining email social networks," in *MSR '06: Proceedings of the 2006 international workshop on Mining software repositories*. New York, NY, USA: ACM, 2006, pp. 137–143.

[2] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan, "Mining email social networks in postgres," in *MSR '06: Proceedings of the 2006 international workshop on Mining software repositories*. New York, NY, USA: ACM, 2006, pp. 185–186.

[3] V. R. de Carvalho and W. W. Cohen, "Learning to extract signature and reply lines from email," in *CEAS*, 2004.

[4] N. Freed and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types," RFC 2046 (Draft Standard), Nov. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2046.txt

[5] Google, "Gmail," http://mail.google.com, 2009, last visited March 2009.

[6] S. Hambridge, "Netiquette Guidelines," RFC 1855 (Informational), 1995. [Online]. Available: http://www.ietf.org/rfc/rfc1855.txt

[7] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "An empirical study of global software development: distance and speed," in *ICSE '01: Proceedings of the 23rd International Conference on Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 81–90.

[8] I. Herraiz, G. Robles, J. J. Amor, T. Romera, and J. M. G. Barahona, "The processes of joining in global distributed software projects," in *GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner*. New York, NY, USA: ACM, 2006, pp. 27–33.

[9] A. Kolcz, A. Chowdhury, and J. Alspector, "Improved robustness of signature-based near-replica detection via lexicon randomization," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2004, pp. 605–610.

[10] M.-L. Lee, T. W. Ling, H. Lu, and Y. T. Ko, "Cleansing data for mining and warehousing," in *DEXA*, ser. Lecture Notes in Computer Science, T. J. M. Bench-Capon, G. Soda, and A. M. Tjoa, Eds., vol. 1677. Springer, 1999, pp. 751–760.

[11] Microsoft, "Microsoft outlook 2007," http://www.microsoft.com/outlook/, 2009, last visited March 2009.

[12] P. C. Rigby and A. E. Hassan, "What Can OSS Mailing Lists Tell Us? A Preliminary Psychometric Text Analysis of the Apache Developer Mailing List," in *MSR '07: Proceedings of the Fourth International Workshop on Mining Software Repositories*. Washington, DC, USA: IEEE Computer Society, 2007, p. 23.

[13] G. Robles and J. M. Gonzalez-Barahona, "Developer identification methods for integrated data from various sources," *SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–5, 2005.

[14] G. Robles, J. M. González-Barahona, D. Izquierdo-Cortazar, and I. Herraiz, "Tools for the study of the usual data sources found in libre software projects," *International Journal of Open Source Software and Processes*, vol. 1, no. 1, pp. 24–45, 2009.

[15] J. Tang, H. Li, Y. Cao, and Z. Tang, "Email data cleaning," in *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA: ACM, 2005, pp. 489–498.

[16] D. C. Čubranić and G. C. Murphy, "Hipikat: recommending pertinent software development artifacts," in *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 408–418.

[17] P. Weissgerber, D. Neu, and S. Diehl, "Small patches get in!" in *MSR '08: Proceedings of the 2008 international working conference on Mining software repositories*. New York, NY, USA: ACM, 2008, pp. 67–76.

[18] K. Whistler, M. Davis, and A. Freytag, "The unicode character encoding model," The Unicode Consortium, Tech. Rep. 17, November 2008.