# Crawling and Preprocessing Mailing Lists At Scale for Dialog Analysis

**Janek Bevendorff** * **Khalid Al-Khatib*** **Martin Potthast**‡ **Benno Stein***
*Bauhaus-Universität Weimar ‡Leipzig University
<first>.<last>@uni-{weimar, leipzig}.de

## Abstract

This paper introduces the Webis Gmane Email Corpus 2019, the largest publicly available and fully preprocessed email corpus to date. We crawled more than 153 million emails from 14,699 mailing lists and segmented them into semantically consistent components using a new neural segmentation model. With 96% accuracy on 15 classes of email segments, our model achieves state-of-the-art performance while being more efficient to train than previous ones. All data, code, and trained models are made freely available alongside the paper.[1]

## 1 Introduction

Email is perhaps the most reliable and ubiquitous means of digital communication. Notwithstanding the mainstream adoption of social media for private communication as of about 2010, email prevails unrivaled for workplace communication and beyond. Compared to social media, however, emails have attracted much less research attention in the fields of computational linguistics, natural language processing, and information retrieval. Key reasons for the neglect can be found in the presumed difficulty of obtaining emails at scale, the lack of open technologies to parse them, and that, despite their importance, they are hardly considered *en vogue*.

Although mailing lists as a rich and accessible source for emails have been tapped before, this has never been done at scale. Our contributions in this respect are (1) the Webis Gmane Email Crawl 2019, a crawl of more than 153 million emails from a wide range of mailing lists, (2) the Chipmunk email segmenter, a newly developed end-to-end neural model, and (3) the complete preprocessing of the crawled emails using our model to construct the largest corpus of "ready-to-use" emails to date. Our corpus encompasses more than 20 years worth of discussions on a diverse set of topics, including important political and societal issues.

[1] https://webis.de/publications.html?q=ACL+2020

We believe that providing the research community with access to clean and preprocessed communication data from emails will foster open research in several areas, such as the analysis of dialogs and discourse, stylometry, language evolution, argument mining, as well as information retrieval, and the synthesis of conversations and argumentation.

## 2 Related Work

For research purposes, the three primary sources of email data are public mailing lists and newsgroups, volunteered or leaked private email datasets, and email databases at companies and service providers. The WestburyLab USENET corpus (Shaoul and Westbury, 2009, 2013) was crawled between 2005 and 2011. More widely employed has been the "20 newsgroups" corpus (Lang, 1995). The W3C corpus compiles the public W3C mailing lists (Wu, 2005), Jiang et al. (2013) examined 8 years of patch submissions to the Linux Kernel Mailing List, and Niedermayer et al. (2017) inspected the process of standardization across IETF bodies via its mailing lists. The CSpace corpus consists of 15,000 student dialogs volunteered for research during a management course at CMU (Kraut et al., 2004).

All of the above have been extensively analyzed (Minkov et al., 2005, 2006; Lawson et al., 2010), yet the most widely studied corpus remains the leaked Enron corpus (Klimt and Yang, 2004), built as part of the U.S. FERC's investigation into the Enron Corporation. It has been subject to studies on speech act and dialog analysis (Goldstein et al., 2006), named entities (Lawson et al., 2010), and word usage patterns (Keila and Skillicorn, 2005), among many others. Another recently leaked dataset comprises the Clinton emails that surfaced during the 2016 U.S. presidential election (De Felice and Garretson, 2018). Regarding email data at companies and service providers, not many researchers are able to disclose their datasets (Avigdor-Elgrabli et al., 2018).



*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1151–1158
July 5 - 10, 2020. 

Regardless of their source, emails are usually unstructured and difficult to process even for human readers (Sobotta, 2016). Thus, many approaches have been proposed for cleansing newsgroup and email data. As one of the earliest, de Carvalho and Cohen (2004) developed a specialized method for detecting and removing signatures based on typical text indicators. Tang et al. (2005) developed a high-accuracy model for detecting blocks of non-content in emails using a mixture of SVM models and hard-coded rules. An unsupervised approach was employed by Contractor et al. (2010), who applied a noisy channel model for filtering out non-content. Similarly, Bettenburg et al. (2011) used spell checking techniques for uncovering technical artifacts like source code, disentangling them from the main content. A more general approach, befittingly named *Zebra*, was published by Lampert et al. (2009), who split messages into a series of structural and semantic "zones", such as *author text* and *signature*. Finally, Repke and Krestel (2018) developed *Quagga*, the first neural end-to-end model inspired by Lampert et al.'s Zebra, which showed very substantial performance improvements. Most machine learning-based approaches rely on classifying lines of text, either by detecting the start and the end of structural blocks with specialized models, or by assessing each line individually via its surrounding context.

With the increase in machine-generated emails, recent studies have shifted their focus away from dialogs and towards parsing and categorizing (Aberdeen et al., 2010; Zhang et al., 2017) or threading notifications (Ailon et al., 2013), as well as automated template induction (Proskurnia et al., 2017; Castro et al., 2018; Kocayusufoglu et al., 2019).

## 3 The Webis Gmane Email Corpus 2019

Our dataset was crawled from Gmane,[2] a popular email-to-newsgroup gateway, which allows users to subscribe to mailing lists via the NNTP newsgroup protocol that formed the basis for the Usenet. While Gmane's web portal has been offline for years and was recently replaced by a minimal website under a new domain name, the newsgroup portal is still alive and messages from active mailing lists arrive every day. Unlike a mailing list server, a newsgroup server keeps an archive of messages, allowing a user to download the history of a newsgroup even if they did not participate in it from the beginning. Traditional newsgroup servers often have a limited retention period, though fortunately, Gmane archived all messages since its launch in 2002. About a million messages date back even further to the year 2000 and a small number even to the early 90's. The latest message in our corpus is from mid-May 2019, which is when we stopped crawling. Considering this enormous time span and the uncertain future of Gmane, we see archiving these messages as both a great research opportunity and an attempt at preserving our digital heritage.

Following the style of the Usenet, Gmane groups are ordered in a hierarchy of subjects under the common *gmane* root. This hierarchy makes it easy to categorize mailing lists into topical domains giving a rough overview of what is being talked about. The majority of groups is of a generally technical nature (e.g., in *gmane.comp* or *gmane.linux*), a large number of other categories exists, most notably *culture*, *politics*, *science*, *education*, *music*, *games*, and *recreation*. Below these main categories, a plethora of individual subjects are found. A cursory topic modeling study reveals not only software development discussions, but also debates about environmental issues, climate change, gender equality, mobility, health, business, international conflicts, general political concerns, philosophy, religious beliefs, and many more.

### 3.1 Acquisition

We crawled all 14,699 groups of which 64 turned out empty. Gmane provides another 18,450 groups under the *gwene* hierarchy for headlines and snippets from RSS feeds. We crawled those as well, but have not analyzed nor added them to the dataset. The crawling process ran slowly over a period of months, producing 604 GiB of compressed WARC files. The total number of messages across all groups sums up to 153,310,330 usable mails. The largest individual group is the Linux Kernel Mailing List with 2.4 million messages followed by the KDE bug tracking list with 2 million. Excluding any obvious bug tracking or software patch submission lists, 113 million messages remain. Further excluding the largest hierarchies *comp*, *linux*, and *os*, 24 million messages are left, which boil down to 7.8 million when restricted to the seven exemplary hierarchies mentioned above. 6.4 million of these are English-language, the rest is mostly German, French, and Spanish. The 153 million messages were posted by 6.4 million unique sender addresses

[2] https://news.gmane.io or rather: nntp://news.gmane.io

and the influx volume amounts to over 710,000 messages per month. This number is a bit lower at 610,000 when only considering the past five years. The top 10 groups account for an average of 1.2 million messages each and the top 10,000 groups for 15,250, while the bottom 5,000 groups have on average 100 messages.

### 3.2 Preprocessing

Emails are a noisy data source in need of heavy preprocessing. The Usenet and early-day mailing lists developed (n)etiquettes for how to write proper messages. These included quoting as little as possible, replying inline, separating signatures by two hyphens, and restricting their length to four lines. Email—the more recent in particular—obeys none of those. For the most part, messages consist of large blocks of nested quotations—often mutilated by the 78-character limit, various formats for introducing quotations, exuberant unstructured personal signatures, and automated signatures added by the author's user agent or the mailing list server. Moreover, technical emails often contain fragments of source code, log data, or diffs. Automated emails also contain semi-structured templates like ASCII-formatted tables. Extracting the content of such unstructured messages proves difficult and long threads pose a challenge even to human readers.

We started the preprocessing by parsing the MIME contents into pure plaintext. To preserve the privacy of users, the name parts of email addresses were replaced with a 16-byte base64 prefix of the address's SHA-256 hashes with *@example.com* appended as the authority part. Headers were reduced to the set necessary for retaining date-time, subject, thread, sender, and recipient information. Finally, the contents of each email were segmented and annotated using our model described in Section 4, allowing for easy extraction of not only the main content, but also other structured information. The final corpus is packaged as compressed line-based JSON files that can be easily indexed into Elasticsearch using its bulk API.

## 4 The Chipmunk Email Segmenter

Cleansing email plaintexts is laborious and first requires splitting them into different functional and semantic segments (also sometimes called zones). Our first attempt at this was a re-implementation of the classic approach by Tang et al. Despite our best efforts, its handcrafted feature set, and the need to train two individual SVMs for each type of content block caused generalizability and scalability issues on our much larger and more diverse dataset. Also, a context window of three lines was not nearly enough to reliably identify all types of content blocks, and making the window larger did not yield satisfying results due to the simplicity and the lack of shared weights among the individual models. We also needed a much more fine-grained segmentation, which not even the more recent neural approach by Repke and Krestel could deliver without substantial changes, so it was decided to develop a new email segmenter.

We identified 15 common segments recurring in emails: (1) *paragraphs* (main content), (2) *salutations*, (3) *closings*, (4) *quotations*, (5) *quotation markers* (quotation author and date), (6) *inline email headers*, (7) *personal signatures*, (8) *automated MUA signatures* (i.e., mail user agent, but also mailing list details or advertising), (9) *source code*, (10) *source code diffs*, (11) *log data*, and (12) *technical noise* (e.g., inline attachments or PGP signatures), (13) semi-structured *tabular data*, (14) *ornaments* (e.g., separator lines), and (15) structural *section headings* (e.g., in a call for papers). We annotated segments in a stratified sample of 3,033 emails from a range of different groups, totaling 170,309 line annotations. Annotated segments are mostly unambiguous so that a single annotator can produce consistent and high-quality annotations in multiple correction passes. Although the sample is technically multilingual, most emails are in English. Of the 3,033 emails, we set aside 300 for model validation and extracted another sample of 1.5 million emails and concatenated them to a single file of 80 million lines (2.8 GiB). Here we replaced all email addresses with the token `@EMAIL@`, all URLs with `@URL@`, mapped numbers to the digit 0, replaced all hexadecimal values with `@HASH@`, runs of four or more indenting spaces with `@INDENT@`, split words on special characters (mainly for tokenizing quotations and source code), and normalized Unicode characters to NFKC. We used this processed dump to train a fastText embedding (Grave et al., 2017) with a default vector dimension of 100.

### 4.1 Model Architecture

The segmentation model has a hybrid RNN-CNN architecture as depicted in Figure 1. For each line, we define a context window of $c = 4$ lines before

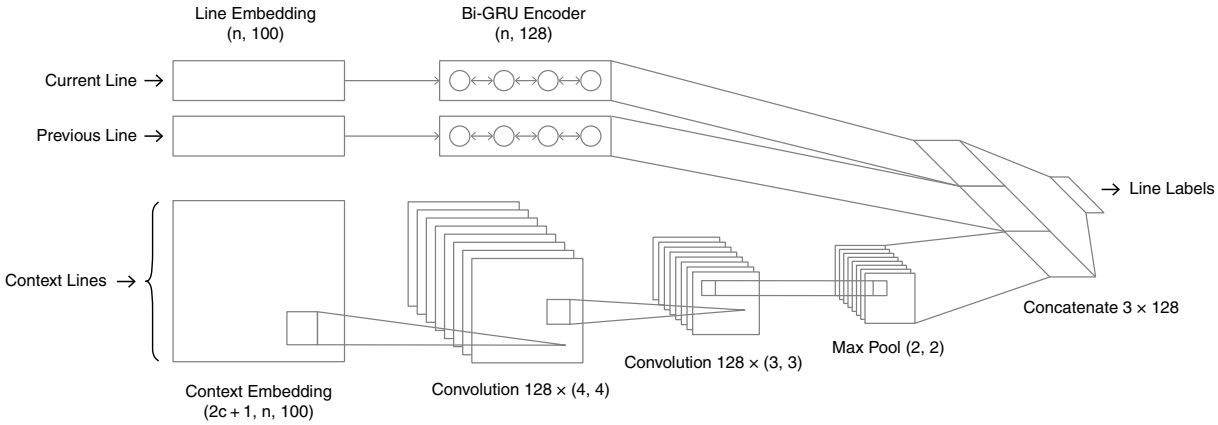


Figure 1: Architecture of the Chipmunk email segmenter. Embeddings for the current and previous lines (max length $n = 12$ words) and a 2D line context window ($c = 4$) are fed into separate inputs. We use batch normalization after the RNN and the first CNN layer and a dropout chance of 0.25 before the final softmax layer.

and after the current line and build an embedding matrix of dimensions $(2c + 1, n, 100)$, $n$ being the maximum word token count per line. Longer lines are truncated by discarding tokens between the first 75% and the last 25% of the line preserving both line beginnings and endings with preference to beginnings, where more structural markers are found under left-to-right writing. Shorter lines and the top or the bottom of the context matrix are padded if required. We feed the line embeddings into separate 128-unit Bi-GRU encoders and the context matrix into a 2D CNN. The idea is that, unlike normal text, plaintext emails have a spatial layout where the horizontal and the vertical axis both convey structural information (most importantly the first column). The CNN performs 128 convolutions with a filter size of $4 \times 4$, then another 128 convolutions with a filter size of $3 \times 3$, and finally a max pooling of $2 \times 2$. After either of the Bi-GRUs and the first convolution, we add in a batch normalization. The CNN output is fed into a 128-dimensional dense layer, concatenated with the other outputs, and then regularized with a dropout of 0.25 before being passed to the softmax layer with outputs for the 15 segment labels and `<empty>` for blank lines. All layers have ReLU as their activation function. We train the model using a mini-batch size of 128 and the Adam optimizer with hinge loss. Choosing this over crossentropy is a decent trade-off between accuracy and generalizability. While crossentropy tends to find a closer fit, giving higher accuracy on very similar data, this comes at the expense of uncertain decisions and early overfitting. Hinge loss prefers larger margins, generalizing better to new and entirely unseen data in a line-wise classification scenario with strict block boundaries.

### 4.2 Evaluation

To evaluate our model, we compare it with two others from the literature in two different settings. Table 1 compiles an overview of the evaluation results. A confusion matrix for our model is found in Table 2 in the appendix. Our model achieves 96% accuracy over all classes. Mapped to binary decisions between paragraphs and non-paragraphs, the accuracy goes up to 98%. The recall on the paragraph class is 93% (see Table 2). The majority class are quotations with 33%, followed by patches with 16%. Paragraphs come in at 11%. Note that the patch class is overrepresented not because we sampled primarily patch emails, but because patches tend to be longer than normal emails. Still, we achieve an overall high accuracy on all classes. A typical segmentation is provided as an example in Figure 3.

To test the model's ability to generalize to unseen data, we annotated 300 emails from the Enron corpus, whose class distribution differs significantly from mailing lists: The emails are much shorter and most lines belong to paragraphs (36%) or empty lines (26%). Quotations account for 8% and code or patches are non-existent. Though significantly lower, our model still shows an acceptable accuracy of about 88%. The excessive use of inline headers containing multiple lines of forwarding addresses appears to be the main challenge for our model, which is expected considering that forwarding emails to dozens of recipients is rare on mailing lists. Furthermore, the proprietary Enron mail user agent had an unusual forwarding and quotation style quite unlike the more common Thunderbird, GMail, or Outlook notations.

Finally, we compared our model against *Quagga*, the state-of-the-art neural segmentation model by Repke and Krestel and a re-implementation of Tang et al.'s SVM email cleaning approach. Unfortunately, a training routine was missing from Quagga's source code, so we re-implemented this part as closely to the original as possible with one notable exception. We changed the way the model handles quotations. The original model did not have a quotation class and was instead trained to ignore quotation indicators so as to predict normal content segments within quotations also. This is very different from how our model handles quotations and it renders the reconstruction of a conversation from the segments alone impossible. We prefer our approach to classify quotations as a separate segment, which retains the structure of emails and one can simply strip the quotation indicators and then apply the model recursively. We trained our own Quagga on all 16 classes for 20 epochs (the model started overfitting after more epochs). Although the original model was trained and tested on only five classes, the extended and retrained model performs only slightly worse than ours with 94% accuracy overall and very similar scores for most of the frequent classes. The degradation on the Enron corpus appears to be worse than in our model (with the exception of the log data class). In conclusion, we can say that both models perform equally well, though our model achieves overall better generalization. In terms of training speed, we found our approach to be faster and more efficient, since it relies on a 2D context window instead of a vertical RNN for sequences of lines.

The model by Tang et al. required a great deal of feature engineering and the training of many separate models. For simplicity, and in accordance with the original paper, we mapped all labels to the reduced set of *content*, *quotation*, *header*, *signature*, *code* (*patch*), and <*empty*>. Despite the smaller number of classes, the model's accuracy lags behind the neural models with 80% on Gmane and only 72% on the Enron corpus.

## 5 Ethical Considerations

The distribution of email data raises ethical concerns, such as possible violations of privacy and legal requirements, which we addressed to the best of our ability. All emails in our corpus are from public mailing lists and by policy, Gmane only accepts such lists whose users are comfortable with their emails being publicly readable. At the time of writing, the original messages in our corpus are openly available to anyone through the NNTP interface and other mailing list archives. Nevertheless, we took measures to avoid abuse of the readily parsed and compiled form of the data, one being the aforementioned anonymization of email addresses to inhibit trivial mass harvesting. Furthermore, we enforce a strict release policy in compliance with the GDPR academic exemptions. Access to the data is granted solely to researchers and academic institutions and we prohibit further distribution for non-academic purposes.

| | Gmane Corpus | | | Enron Corpus | | |
|---|---|---|---|---|---|---|
| | *Ours* | Quagga | Tang | *Ours* | Quagga | Tang |
| All Classes | **0.96** | 0.94 | 0.80 | **0.88** | 0.83 | 0.72 |
| Quotation | **0.99** | **0.99** | **0.99** | **0.99** | 0.88 | 0.85 |
| Patch | **0.95** | **0.95** | 0.46 | – | – | – |
| Paragraph | **0.93** | 0.90 | 0.90 | **0.95** | 0.91 | 0.89 |
| Log Data | **0.84** | 0.77 | – | 0.24 | **0.74** | – |
| MUA Sig. | 0.91 | **0.93** | 0.4 | **0.65** | 0.51 | 0.21 |
| Personal Sig. | 0.77 | **0.85** | | **0.85** | 0.78 | |

Table 1: Segmentation performance of the Chipmunk model compared to Quagga by Repke and Krestel and Tang et al.'s SVM approach. We report overall categorical accuracy and recall for the six most frequent classes, excluding empty lines. The models were run on the Gmane corpus and a small annotated subset of the Enron corpus to analyze domain transfer.

## 6 Summary

This paper contributes the largest email corpus to date. The corpus is targeted mainly at discussion and dialog-based research in NLP. We gave an overview of the topics discussed in the corpus, demonstrating that it is a valuable source for several NLP tasks, such as argument mining. Despite the prevalence of technical conversations, various important and controversial societal issues are covered in the corpus as well. To minimize user overhead, we developed a new neural model for segmenting emails with high precision and recall, which achieves state-of-the-art performance, allowing for fine-grained extraction of structural elements from emails. All the resources developed in this paper are freely available.[3]

[3]Visit https://webis.de/data.html?q=Webis-Gmane-19 for details about gaining access to the corpus. The pre-trained Chipmunk model as well as the code we used for training it and for conducting our experiments are hosted at GitHub (https://github.com/webis-de/ACL-20).

## References


Douglas Aberdeen, Ondrey Pacovsky, and Andrew Slater. 2010. The learning behind gmail priority inbox. In *LCCC : NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds*.

Nir Ailon, Zohar S Karnin, Edo Liberty, and Yoelle Maarek. 2013. Threading machine generated email. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 405–414.

Noa Avigdor-Elgrabli, Roei Gelbhart, Irena Grabovitch-Zuyev, and Ariel Raviv. 2018. More than threads: Identifying related email messages. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1711–1714.

Nicolas Bettenburg, Bram Adams, Ahmed E Hassan, and Michel Smidt. 2011. A lightweight approach to uncover technical artifacts in unstructured data. In *2011 IEEE 19th international conference on program comprehension*, pages 185–188.

Vitor Rocha de Carvalho and William W. Cohen. 2004. Learning to extract signature and reply lines from email. In *CEAS 2004 - First Conference on Email and Anti-Spam, July 30-31, 2004, Mountain View, California, USA*.

Dotan Di Castro, Iftah Gamzu, Irena Grabovitch-Zuyev, Liane Lewin-Eytan, Abhinav Pundir, Nil Ratan Sahoo, and Michael Viderman. 2018. Automated extractions for machine generated mail. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 655–662.

Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. 2010. Unsupervised cleansing of noisy text. In *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 189–196.

Rachele De Felice and Gregory Garretson. 2018. Politeness at work in the Clinton email corpus: A first look at the effects of status and gender. *Corpus Pragmatics*, 2(3):221–242.

Jade Goldstein, Andres Kwasinksi, Paul R. Kingsbury, Roberta Evans Sabin, and Albert McDowell. 2006. Annotating subsets of the Enron email corpus. In *CEAS 2006 - The Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California, USA*.

Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431.

Yujuan Jiang, Bram Adams, and Daniel M. Germán. 2013. Will my patch make it? and how fast?: case study on the Linux kernel. In *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, San Francisco, CA, USA, May 18-19, 2013*, pages 101–110.

P. S. Keila and David B. Skillicorn. 2005. Structure in the Enron email dataset. *Computational & Mathematical Organization Theory*, 11(3):183–199.

Bryan Klimt and Yiming Yang. 2004. The Enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226.

Furkan Kocayusufoglu, Ying Sheng, Nguyen Vo, James Bradley Wendt, Qi Zhao, Sandeep Tata, and Marc Najork. 2019. Riser: Learning better representations for richly structured emails. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 886–895.

Robert Kraut, S. Fussell, F. Lerch, and J. Espinosa. 2004. Coordination in teams: evidence from a simulated management game.

Andrew Lampert, Robert Dale, and Cécile Paris. 2009. Segmenting email message text into zones. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 919–928.

Ken Lang. 1995. Newsweeder: learning to filter Netnews. In *Proceedings of ICML*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

Nolan Lawson, Kevin Eustice, Mike Perkowitz, and Meliha Yetisgen-Yildiz. 2010. Annotating large email datasets for named entity recognition with mechanical turk. In *Proceedings of the 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, Los Angeles, USA, June 6, 2010*, pages 71–79.

Einat Minkov, William W. Cohen, and Andrew Y. Ng. 2006. Contextual search and name disambiguation in email using graphs. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 27–34.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names from email: Applying named entity recognition to informal text. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*, pages 443–450.

Heiko Niedermayer, Nikolai Schwellnus, Daniel Raumer, Edwin Cordeiro, and Georg Carle. 2017. Information mining from public mailing lists: A case study on IETF mailing lists. In *Internet Science - 4th International Conference, INSCI 2017, Thessaloniki, Greece, November 22-24, 2017, Proceedings*, pages 301–309.

Julia Proskurnia, Marc-Allen Cartright, Lluis Garcia Pueyo, Ivo Krka, James Bradley Wendt, Tobias Kaufmann, and Balint Miklos. 2017. Template induction over unstructured email corpora. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1521–1530.

Tim Repke and Ralf Krestel. 2018. Bringing back structure to free text email conversations with recurrent neural networks. In *Proceedings of Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018*, volume 10772 of *Lecture Notes in Computer Science*, pages 114–126.

Cyrus Shaoul and Chris Westbury. 2009. A USENET corpus (2005–2009). *University of Alberta, Canada*.

Cyrus Shaoul and Chris Westbury. 2013. A reduced redundancy USENET corpus (2005–2011). *University of Alberta*.

Nikolai Sobotta. 2016. Why forwarded email threads are hard to read: The email format as an antecedent of email overload. *CAIS*, 39:2.

Jie Tang, Hang Li, Yunbo Cao, and ZhaoHui Tang. 2005. Email data cleaning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 489–498.

Yejun Wu. 2005. Email messages corpus parsed from W3C lists (for TRECENT 2005).

Aston Zhang, Lluis Garcia Pueyo, James Bradley Wendt, Marc Najork, and Andrei Z. Broder. 2017. Email category prediction. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 495–503.

# 7 Appendix

## 7.1 Segmentation Confusion Matrix

| | par | clos | ihead | log | msig | pat | psig | quot | mark | code | salu | head | tab | tech | sep | emp |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| par | 0.933 | 0.006 | 0.000 | 0.014 | 0.007 | 0.003 | 0.006 | 0.008 | 0.001 | 0.004 | 0.001 | 0.001 | 0.012 | 0.000 | 0.001 | 0.002 |
| clos | 0.051 | 0.907 | 0.000 | 0.000 | 0.000 | 0.000 | 0.009 | 0.014 | 0.000 | 0.000 | 0.005 | 0.000 | 0.005 | 0.000 | 0.005 | 0.005 |
| ihead | 0.000 | 0.000 | 0.915 | 0.000 | 0.007 | 0.000 | 0.007 | 0.007 | 0.000 | 0.000 | 0.000 | 0.000 | 0.065 | 0.000 | 0.000 | 0.000 |
| log | 0.074 | 0.001 | 0.000 | 0.843 | 0.001 | 0.028 | 0.000 | 0.000 | 0.000 | 0.010 | 0.000 | 0.001 | 0.026 | 0.000 | 0.009 | 0.007 |
| msig | 0.047 | 0.000 | 0.000 | 0.000 | 0.914 | 0.020 | 0.010 | 0.003 | 0.000 | 0.000 | 0.000 | 0.003 | 0.001 | 0.001 | 0.000 | 0.000 |
| pat | 0.014 | 0.000 | 0.000 | 0.005 | 0.000 | 0.948 | 0.000 | 0.000 | 0.001 | 0.022 | 0.000 | 0.000 | 0.004 | 0.000 | 0.002 | 0.004 |
| psig | 0.068 | 0.045 | 0.000 | 0.019 | 0.060 | 0.000 | 0.774 | 0.019 | 0.000 | 0.000 | 0.000 | 0.000 | 0.008 | 0.000 | 0.008 | 0.000 |
| quot | 0.007 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.991 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 |
| mark | 0.019 | 0.000 | 0.005 | 0.000 | 0.000 | 0.005 | 0.000 | 0.000 | 0.947 | 0.000 | 0.005 | 0.000 | 0.005 | 0.000 | 0.005 | 0.010 |
| code | 0.107 | 0.000 | 0.000 | 0.231 | 0.000 | 0.030 | 0.000 | 0.006 | 0.000 | 0.621 | 0.000 | 0.000 | 0.006 | 0.000 | 0.000 | 0.000 |
| salu | 0.017 | 0.017 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.950 | 0.000 | 0.017 | 0.000 | 0.000 | 0.000 |
| head | 0.312 | 0.062 | 0.000 | 0.062 | 0.125 | 0.000 | 0.000 | 0.188 | 0.000 | 0.000 | 0.000 | 0.125 | 0.125 | 0.000 | 0.000 | 0.000 |
| tab | 0.085 | 0.000 | 0.009 | 0.030 | 0.004 | 0.026 | 0.000 | 0.102 | 0.000 | 0.000 | 0.000 | 0.000 | 0.728 | 0.000 | 0.004 | 0.013 |
| tech | 0.077 | 0.000 | 0.000 | 0.000 | 0.038 | 0.000 | 0.000 | 0.000 | 0.038 | 0.038 | 0.000 | 0.000 | 0.000 | 0.731 | 0.038 | 0.038 |
| sep | 0.008 | 0.003 | 0.000 | 0.008 | 0.000 | 0.005 | 0.008 | 0.023 | 0.005 | 0.000 | 0.000 | 0.000 | 0.003 | 0.000 | 0.938 | 0.000 |
| emp | 0.005 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.995 |

Table 2: True labels are on the vertical axis, values were normalized line-wise. Classes: *paragraph*, *closing*, *inline_headers*, *log_data*, *mua_signature*, *patch*, *personal_signature*, *quotation*, *quotation_marker*, *raw_code*, *salutation*, *section_heading*, *tabular*, *technical*, *visual_separator*, <*empty*>. The model is generally conservative, leaning towards paragraphs in uncertain cases. A slight yet notable confusion between MUA signatures, personal signatures, and closings can be observed, which are sometimes hard to discern even for humans. The heading class is the least prevalent of all and thus missing training data. Empty line misclassification is corrected afterwards.

## 7.2 Corpus Statistics

| | Languages | | | | |
|---|---|---|---|---|---|
| | **All** | **EN** | **DE** | **FR** | **n/a** |
| Messages | 153.3M | 137.8M | 1.9M | 1.8M | 2.1M |
| Excl. Replies | 57.1M | 51.8M | 513.4k | 683.9k | 1.1M |
| Excl. Bugs, Patches | 113.2M | 100.3M | 1.9M | 1.6M | 806.0k |
| Excl. comp, linux, os | 24.0M | 19.3M | 448.5k | 315.6k | 172.8k |
| Messages/Month | 710.6k | 640.8k | 9,0k | 8,5k | 10.0k |
| Unique Groups | 14,635 | 14,398 | 6,984 | 7,710 | 9,241 |
| Unique Senders | 6.4M | 6.7M | 164.6k | 137.7k | 252.3k |
| Paragraph Lines | 2.0G | 1.8G | 28.5M | 28.7M | 4.7M |
| Quotation Lines | 2.5G | 2.3G | 26.0M | 26.5M | 5.0M |
| MUA Sig. Lines | 400.3M | 347.9M | 3.8M | 3.5M | 6.3M |
| Pers. Sig. Lines | 158.9M | 133.3M | 5.3M | 2.3M | 396.6k |
| Patch Lines | 2.0G | 1.9G | 7.4M | 18.5M | 7.7M |
| Code Lines | 254.0M | 235.0M | 1.6M | 2.2M | 339.9k |

Table 3: Gmane corpus statistics by detected language.

## 7.3 Segmentation Examples

**head:** CALENDAR ENTRY: APPOINTMENT

**tab:** Description:
**tab:** EB48c2 - DPR Risk Mtg.
**tab:** Date 3/6/2001
**tab:** Time 10:00 AM - 11:00 AM (Central Standard Time)
**tab:** Chairperson: Stacey W White

**head:** Detailed Description:

**par:** Shona Wilson Heading the meeting

Figure 2: Segmentation example of an Enron email with section headings, tabular data, and a paragraph.

**salu:** Hi Michael,

**par:** Thanx very much for your response to my question. I will keep a look
**par:** out on VITN for any updates. The artwork has been fantastic over the
**par:** years! Thanx so much for all the effort put in!!!

**clos:** kind regards
**clos:** LiveMiles

**msig:** Sent from my iPhone

**mark:** On Apr 8, 2010, at 20:52, "michael_. . . " </hOMQPRV. . . @example.com
**mark:** > wrote:
**. . .**
**quot:** > > Sent from my iPhone
**quot:** > >
**quot:** > Hi Miles & others,
**quot:** >
**quot:** > sorry for the late reply. In September last year I have published
**quot:** > new artwork for TT I and TL VII by Leah Cim on Voices In The Net.
**quot:** > There will be an update of the site quite soon (I hope), featuring
**. . .**

**tech:** [Non-text portions of this message have been removed]

**sep:** ————————————

**sep:** ————————————

**msig:** http://www.tadream.net
**msig:** ————————————Yahoo! Groups Links

**msig:** <*> To visit your group on the web, go to:
**msig:** http://groups.yahoo.com/group/tadream/

**msig:** <*> Your email settings:
**msig:** Individual Email | Traditional

**msig:** <*> To change settings online go to:
**msig:** http://groups.yahoo.com/group/tadream/join
**msig:** (Yahoo! ID required)
**. . .**

Figure 3: Gmane corpus email segmentation example. Lines were identified correctly as salutation, paragraph, closing, MUA signatures, quotation marker, quotation, technical, and separators.