A Replication Study of Communication Style Analysis in the Linux Kernel Mailing List

Júlia Azevedo
PUC-Rio/INSA/Inria
Rio de Janeiro, Brazil
juliatadeu@aise.inf.puc-rio.br

Theo Canuto
PUC-Rio
Rio de Janeiro, Brazil
theo@aise.inf.puc-rio.br

Heraldo Borges
INSA/Inria/IRISA
Rennes, France
heraldo.pimenta-borges-filho@irisa.fr

Mathieu Acher
INSA/Inria/IRISA
Rennes, France
mathieu.acher@irisa.fr

Juliana Alves Pereira
PUC-Rio
Rio de Janeiro, Brazil
juliana@inf.puc-rio.br

Abstract

The Linux Kernel Mailing List (LKML) has long served as an important empirical source for studying communication dynamics in large-scale open-source software development. However, replicating historical LKML studies under contemporary data conditions remains challenging due to differences in archival infrastructures, metadata consistency, and dataset construction practices. This paper presents a qualitative replication of the study *Differentiating Communication Styles of Leaders on the Linux Kernel Mailing List* using the LKML5Ws curated dataset. We first investigate the feasibility of reconstructing LKML corpora directly from official archives using the *lei* interface and discuss the practical limitations encountered during large-scale data acquisition. We then re-execute the original analytical pipeline using LKML5Ws, maintaining high fidelity to the original methodological assumptions and execution environment. Through this replicated analysis, we evaluate multiple dimensions, including authorship classification, lexical statistics, expletive usage, and part-of-speech distributions. Ultimately, this work explores both the feasibility and the methodological challenges of revisiting established empirical studies using a contemporary LKML dataset.

Keywords

Linux Kernel Mailing List, empirical software engineering, replication study, communication analysis

1 Introduction

The Linux Kernel Mailing List (LKML) is one of the largest and longest-running socio-technical communication platforms in open-source software development [15]. It serves as the primary coordination channel for Linux kernel development, supporting discussions about architecture decisions, performance trade-offs, bug fixing, and long-term maintenance. Due to its scale and centrality in the Linux ecosystem, LKML has become an important empirical source for studying communication practices, collaboration dynamics, and decision-making processes in large software projects [10, 12].

Prior research has shown that communication in LKML reflects not only technical concerns, but also social and organizational dynamics within the kernel community. In particular, studies on developer interactions have investigated conflict, fairness, leadership, and discourse style in open-source development environments [7, 9, 13]. Among these studies, *Differentiating Communication Styles of Leaders on the Linux Kernel Mailing List* [15] remains a widely referenced work. The study demonstrated that linguistic and stylistic features extracted from mailing list messages can be used to distinguish communication patterns of prominent kernel contributors, particularly Linus Torvalds and Greg Kroah-Hartman.

Although influential, revisiting this type of study under contemporary data conditions presents important methodological challenges. While the original study made its dataset available, our objective was not to reproduce the exact original corpus, but rather to evaluate whether the analytical pipeline could be executed using modern LKML archives and contemporary curated datasets. This introduces practical challenges related to data acquisition, metadata consistency, and corpus comparability across different archival infrastructures. As part of this process, the project initially explored whether a comparable corpus could be reconstructed directly from the official LKML archives hosted at lore.kernel.org [1], following an approach conceptually aligned with the original study.

To this end, we conducted exploratory experiments using *lei*, a command-line interface for querying public-inbox archives. While this approach enabled the retrieval of message metadata and raw email content over limited time ranges, it also revealed substantial challenges related to scalability, message body reconstruction, and thread completeness when extended to the full LKML history. These difficulties highlighted the gap between theoretical data availability and practical dataset construction for longitudinal analyses.

In response to these limitations, we identified and adopted LKML5Ws [11], a curated dataset that was under active development during the course of this study. LKML5Ws provides a large-scale, structured representation of LKML communications, including cleaned message bodies, normalized metadata, and explicit thread information. Using this dataset, we applied the analytical pipeline of the original study [15] under controlled conditions, preserving its core assumptions and execution environment as closely as possible.

The results obtained using LKML5Ws exhibit patterns consistent with the original findings, including clear stylistic differences across prominent kernel contributors and strong discriminative signals in linguistic features. At the same time, the experiments reveal methodological considerations related to corpus composition, metadata completeness, and inherited analytical assumptions, which constrain strict quantitative comparability.

This paper makes three main contributions: (i) An evaluation on the suitability of the LKML5Ws dataset for supporting replication studies on developer communication in LKML. (ii) A qualitative replication of a classic LKML communication study, discussing the feasibility of re-executing its analytical workflow and the limitations imposed by modern data sources. (iii) The availability of all artifacts establishing a foundation basis for future research.

2 Background and Related Work

The Linux Kernel Mailing List (LKML) has long served as an important empirical source for studying communication practices, coordination dynamics, and social interaction in large-scale open-source software development. Prior work has used mailing list archives as socio-technical artifacts to investigate collaboration structures, contributor behavior, leadership dynamics, and discourse patterns within distributed software communities [10, 12].

Among the most influential studies in this area, Schneider et al. [15] examined linguistic and stylistic differences in approximately 46,000 messages authored by Linus Torvalds and Greg Kroah-Hartman. By leveraging textual features such as adverbs, politeness markers, and expletives, the study demonstrated that leadership communication styles can be quantitatively distinguished. Beyond its substantive findings, the work also proposed a clearly defined analytical pipeline combining feature extraction, part-of-speech analysis, and supervised classification techniques. In our study, this work functions as an anchor study, motivating both the research focus and the decision to revisit an established analytical approach under contemporary data conditions.

Subsequent research expanded the investigation of communication phenomena in open-source development environments. Studies have explored conflict, fairness, emotional tone, and incivility in collaborative software discussions, revealing how communication patterns influence technical coordination and contributor experience [6–9, 13, 14]. Research has also investigated communication practices surrounding patch submission and technical discussions in LKML, showing how interaction styles influence review efficiency and developer participation [17]. More recent work has also investigated aggressive and confrontational interactions in LKML, highlighting the continued relevance of discourse analysis in kernel development communities [4].

In parallel, other studies have examined the structural dimensions of LKML data. The DUKS study [12], for example, linked mailing list discussions, commit history, and maintainer information through visualizations, revealing discrepancies between formal roles and contribution behavior. It also highlights the importance of accurate thread reconstruction and temporal metadata for analyzing communication dynamics in large-scale software projects.

Taken together, these studies demonstrate the analytical potential of LKML data for understanding both human and technical aspects of kernel development, while also exposing the practical challenges of dataset construction, documentation, and reuse. In contrast to prior work that either assumes the availability of a suitable dataset or focuses on proposing new analyses, our work explicitly centers on the process of revisiting an established study under modern data constraints. By documenting the difficulties of dataset reconstruction, evaluating a curated alternative, and reassessing an existing analytical pipeline, we aim to clarify the conditions under which communication-based studies of LKML can be reliably reproduced and extended.

3 Study Design

Mailing lists remain vital empirical data sources in software engineering, despite challenges regarding long-term preservation, infrastructure evolution, and dataset reuse [16]. Recent initiatives in computational science also emphasize replication studies and transparent artifacts for improving methodological robustness [3, 14]. Following the definitions proposed by Freire et al. [3], a study is considered reproducible if it provides the original data and code to recreate results, whereas replication involves independently collecting new data to verify findings. Therefore, this work positions itself as a replication rather than a reproduction.

Table 1: Comparison between the original and present study.

Aspect	Original Study [15]	This Study
Data source	LKML archives	Curated LKML5Ws dataset
Time period	Jun 1995 and Feb 2015	Not analyzed
Email aliases	Manually curated	Different coverage and distribution
Temporal analysis	Used	Not used
Analytical pipeline	Original	Preserved
Execution environment	Python 2.x	Python 2.7.18
Goal	Style differentiation	Style differentiation

This study conducts a qualitative replication of *Differentiating Communication Styles of Leaders on the Linux Kernel Mailing List* [15]. Sharing the original analytical focus without proposing new hypotheses, the objective is to revisit the analytical pipeline under contemporary conditions to assess its viability using modern archives and curated datasets.

By framing the original study as an anchor rather than as a benchmark to be improved upon, this work emphasizes methodological transparency over novelty. The contribution lies in documenting the replication process, the challenges of reconstructing datasets, and the feasibility of obtaining meaningful results despite these constraints. This perspective is highly relevant for empirical studies of long-lived software projects navigating evolving data sources, tooling, and archival practices.

We describe the study steps and procedures as follows.

(Step 1) Dataset Selection and Characteristics. This study adopts the LKML5Ws dataset from the Mailing Lists Archiver project. LKML5Ws provides a large-scale, curated representation of Linux kernel communications, aggregating over 20 million emails from 345 lists spanning approximately two decades. The dataset captures both message content and the relational context necessary for studying decentralized software projects.

Replicating the original study required more than just replacing the data source. Because LKML5Ws features a different metadata schema and a substantially altered distribution of contributor aliases and message counts, we had to reconstruct contributor identities through updated alias mappings, select equivalent message fields, and balance classes to preserve methodological fidelity.

Table 1 summarizes the primary differences between the original dataset [15] and LKML5Ws. Notably, a subset of LKML5Ws currently lacks temporal metadata, preventing reliable chronological analyses across the entire corpus, though active development of the dataset may resolve this in the future. Additionally, while the original research manually curated aliases, LKML5Ws's distinct distribution required new corpus balancing decisions to maintain stylistic comparability between contributors.

Despite these data variations, the original analytical pipeline and computational environment - including Python 2.7.18 - were preserved. Ultimately, this highlights our prioritization of methodological fidelity while adapting the data source, reinforcing this work as a replication study evaluating whether previously identified socio-technical communication signals remain observable despite LKML evolution and differing corpus construction.

Table 2: Relevant fields of the LKML5Ws dataset.

Field	Description
from	Email sender identifier
to	List of primary recipients
cc	List of carbon-copy recipients
subject	Email subject line
date	Timestamp indicating when the message was sent
message-id	Unique identifier of the email message
in-reply-to	Message ID of the parent email, if applicable
references	List of message IDs referenced by the email
list	Mailing list where the message was posted
raw_body	Full plaintext representation of the email body
code	Parsed diff-like representations of code fragments
trailers	Parsed signature trailers indicating authorship and attribution

(Step 2) Data Representation and Relevant Fields. LKML5Ws is distributed as an Apache Parquet dataset partitioned by mailing list, enabling scalable and selective data access. Each record represents a single email enriched with structured metadata and parsed content fields. Explicit thread identifiers, complete message bodies, and normalized metadata make LKML5Ws highly suitable for reproducing content-based communication analyses while preserving conversational context. Table 2 summarizes the fields most relevant to our analyses.

As in the original study, authorship identification relies on mapping multiple aliases to a single contributor. Table 3 outlines the email addresses for Linus Torvalds and Greg Kroah-Hartman in LKML5Ws, alongside the retained message counts. Compared to the original study (Table 4), LKML5Ws offers broader coverage and a different alias distribution; notably, Kroah-Hartman is represented by a significantly larger number of messages than Torvalds; Table 3 therefore reports his counts after the downsampling described in Step 3.

Although this corpus does not perfectly match the original message subset, its suitability for replication rests on two factors. First, methodological comparability is maintained by focusing on the same two prominent, stylistically contrasting contributors while preserving the original analytical pipeline and feature extraction methods. Second, sufficient linguistic diversity is ensured by the large, balanced corpus (32,885 messages per author). This diversity is empirically confirmed in our descriptive analyses (Section 4), which demonstrate broad vocabulary usage and high lexical diversity scores of 0.698 and 0.657 for Linus and Greg, respectively.

Table 3: Email aliases and message counts for Linus Torvalds and Greg Kroah-Hartman in our LKML5Ws subset.

Sender	Email	Count
Greg Kroah-Hartman	gregkh@linuxfoundation.org	28753
Greg Kroah-Hartman	gregkh@suse.de	2349
Greg Kroah-Hartman	greg@kroah.com	1765
Greg Kroah-Hartman	greg@wirex.com	8
Greg Kroah-Hartman	gregkh@linux-foundation.org	7
Greg Kroah-Hartman	gregkh@google.com	2
Greg Kroah-Hartman	gregkh@kernel.org	1
Linus Torvalds	torvalds@linux-foundation.org	25119
Linus Torvalds	torvalds@osdl.org	4202
Linus Torvalds	torvalds@transmeta.com	3412
Linus Torvalds	torvalds@linuxfoundation.org	150
Linus Torvalds	linus@torvalds.org	3
Linus Torvalds	linus@linux-foundation.org	1
Linus Torvalds	linus@linuxfoundation.org	1
Linus Torvalds	t0rvalds@transmeta.com	1
Linus Torvalds	tordalds@transmeta.com	1

(Step 3) Analytical Pipeline and Model Training. Our analytical pipeline closely follows the original study. Message bodies extracted from the raw_body field were preprocessed to remove non-linguistic artifacts, such as quoted replies and code fragments. Textual features were then derived from the cleaned content to support descriptive and classification-based analyses.

Table 4: Email aliases and message counts for Linus Torvalds and Greg Kroah-Hartman in the original study [15].

Sender	Email	Count
Greg Kroah-Hartman	greg@kroah.com	13730
Greg Kroah-Hartman	gregkh@suse.de	5179
Greg Kroah-Hartman	gregkh@linuxfoundation.org	3348
Greg Kroah-Hartman	gregkh@xxxxxxxxxxxxxxxxxxx	1632
Greg Kroah-Hartman	greg@xxxxxxxxx	235
Linus Torvalds	torvalds@linux-foundation.org	10365
Linus Torvalds	torvalds@transmeta.com	5959
Linus Torvalds	torvalds@osdl.org	4169
Linus Torvalds	torvalds@xxxxxxxxxxxxxxxxxxx	1139

For authorship differentiation, we employed a Naïve Bayes classifier trained on these extracted features, which included token-based word usage and aggregated linguistic indicators (e.g., part-of-speech frequencies and expletives). Feature extraction and model training strictly adhered to the original methodological assumptions to prioritize pipeline fidelity.

To distinguish between messages authored by Linus Torvalds and Greg Kroah-Hartman, we adopted a data balancing strategy. Because Kroah-Hartman was represented by substantially more messages in LKML5Ws, we randomly downsampled his messages to 32,885 to match Torvalds's count. This avoided distortions from asymmetric representation and preserved stylistic comparability. The downsampling utilized the Python pandas sample function with a fixed random seed (42) to ensure exact reproducibility (script available in supplementary materials[1]).

Model evaluation followed the original Monte Carlo cross-validation strategy: ten executions with random stratified 80/20 training-validation splits. To identify stable communication signals under contemporary LKML conditions, we used the F1-score for classification performance. Further execution details are in Section 4.

This study is guided by three research questions (RQs), as follows:

RQ_1: How accurately can the classifier distinguish between the messages of the two contributors in the LKML5Ws dataset? To determine how accurately the model could distinguish between the two contributors in the LKML5Ws dataset, we evaluated the classifier on a balanced binary corpus comprising 65,775 messages. Class balance was achieved through standard preprocessing and the random downsampling of Greg's messages. The model's performance was then quantitatively assessed using standard classification metrics: average accuracy, precision, recall, and F1-score.

RQ_2: What linguistic features and readability metrics characterize the stylistic differences between the two contributors? To characterize the stylistic differences between the two contributors, we computed a series of descriptive linguistic statistics across the corpus. We extracted normalized metrics focusing on structural elements — average word and sentence counts — as well as lexical diversity to gauge vocabulary usage. Additionally, we applied standard readability formulas, including Flesch Reading Ease and estimated grade-level complexity, to quantitatively compare the linguistic complexity of each contributor's messages.

[1] https://github.com/aisepucrio/LKML-VEM2026/

RQ_3: How do expletive usage, part-of-speech distributions, and discriminative vocabulary differentiate the communication styles of the two contributors? To evaluate specific lexical and grammatical differences between the contributors, our analysis targeted three distinct dimensions. First, we analyzed expletive usage using a lexicon-based string-matching approach. To maintain strict methodological fidelity to the original study, we adopted the same pre-defined dictionary, which was originally derived from the 'Google list of expletives'. A secondary filtering step was applied strictly to exclude known overlapping technical terms and explicit contributor names (e.g., 'wang', 'cox') identified in the corpus.

Rather than expanding the analysis to additional contributors, we retain the original focus on Linus and Greg, whose strong stylistic contrast provides an appropriate baseline for assessing whether the original findings remain reproducible on a contemporary dataset.

4 Results

4.1 (RQ_1) Classification Performance

The classifier was evaluated on a balanced binary corpus containing 65,775 messages after preprocessing and random downsampling of Greg Kroah-Hartman's messages.

The classifier achieved an average accuracy of 92.97%, precision of 96.81%, recall of 88.87%, and F1-score of 92.67%. These results indicate that stylistic differences between the two contributors remain sufficiently distinctive to support robust authorship classification, despite changes in dataset composition and corpus construction.

Although direct numerical comparison with the original study - which reported an F1-score of 96.3% and an accuracy of 96.2% - is limited by differences in corpus coverage and the use of a balanced sampling strategy, the replicated model produces qualitatively consistent signals. In particular, high precision suggests that messages attributed to a given contributor are rarely confused, while the slightly lower recall indicates occasional missed classifications, likely reflecting stylistic overlap and increased corpus diversity in the contemporary dataset.

4.2 (RQ_2) Descriptive Linguistic Statistics

Descriptive linguistic statistics further highlight stylistic differences between the two contributors. Table 5 summarizes normalized metrics for message length, sentence structure, lexical diversity, and readability. The replicated results indicate that Greg Kroah-Hartman writes substantially longer messages on average (524 words versus 279 for Linus Torvalds), despite presenting a slightly lower average sentence count. Linus Torvalds, in contrast, exhibits greater lexical diversity (0.698 versus 0.657), suggesting comparatively more varied vocabulary usage.

Readability metrics reveal particularly strong differences between the two contributors. Messages authored by Linus Torvalds achieve considerably higher readability scores (Flesch Reading Easec[2] of 55.2 versus −0.1), while Greg Kroah-Hartman exhibits substantially higher estimated grade-level complexity (29.1 versus 14.1). Although readability metrics should be interpreted cautiously in technical mailing-list discussions due to domain-specific vocabulary and code-related terminology, these results suggest consistent stylistic differences in writing structure and linguistic complexity.

[2]https://readable.com/readability/flesch-reading-ease-flesch-kincaid-grade-level/

Taken together, these findings remain qualitatively aligned with the original study, supporting the interpretation that distinct communication styles can still be identified under contemporary dataset conditions despite changes in corpus composition.

Table 5: Message content statistics by contributor.

Metric	Linus Torvalds	Greg Kroah-Hartman
Average Word Count	279.0	524.0
Average Sentence Count	10.01	9.09
Lexical Diversity	0.698	0.657
Flesch Reading Ease	55.20	-0.10
Grade Level	14.10	29.10

4.3 (RQ_3) Expletive Usage and Vocabulary

As reported in the original study, the replicated dataset also reveals clear differences in expletive usage between the two contributors. Linus exhibits substantially higher expletive frequency than Greg (4653 against 3041). Although Greg Kroah-Hartman also presents a non-negligible number of detected expletives, closer inspection suggests that part of this vocabulary reflects ambiguities introduced by lexicon-based detection rather than actual offensive language.

The original study reported that Linus Torvalds frequently combines profanity with emphatic criticism in technical discussions. Similar qualitative tendencies remain visible in the replicated corpus, reinforcing the interpretation that expletive usage constitutes a distinguishable stylistic dimension between the two contributors.

Beyond aggregate counts, the distribution of individual expletive terms provides further insight into stylistic differences between the two contributors. Figure 1 presents the most frequent expletives identified in the replicated dataset, together with their relative association with each contributor. The results indicate that expletive usage is concentrated around a relatively small set of recurring terms. Closer inspection, however, reveals methodological limitations inherited from the original expletive detection approach. Because expletives are identified through string-based matching, certain surnames and technical identifiers are incorrectly classified as offensive language. For example, the tokens *"cox"*, *"wang"*, and *"willy"* are frequently associated with Greg Kroah-Hartman despite often referring to contributor names rather than profanity. Similarly, occurrences of the token *"dick"* in Greg's messages commonly refer to contributor names, whereas in Linus' messages the same term appears more frequently in an offensive context. These observations suggest that part of the vocabulary associated with Greg reflects ambiguities introduced by lexicon-based detection rather than genuine expletive use. Although the overall asymmetry in expletive usage remains consistent with the original study, this constitutes an important deviation observed in the replication, highlighting limitations in the inherited lexicon-based detection strategy when applied to a contemporary LKML corpus.

To better isolate stylistic patterns from these ambiguities, Figure 2 presents the most frequent expletive terms after removing tokens identified as contributor-name overlaps and other non-profane ambiguities. Even after this filtering step, the overall pattern remains qualitatively consistent with the original study, with Linus Torvalds exhibiting substantially stronger association with explicit and emphatic language during technical discussions.

Normalized part-of-speech distributions provide additional evidence of stylistic variation between the two contributors. As shown in Figure 3, Linus Torvalds uses adverbs (RB) at a substantially

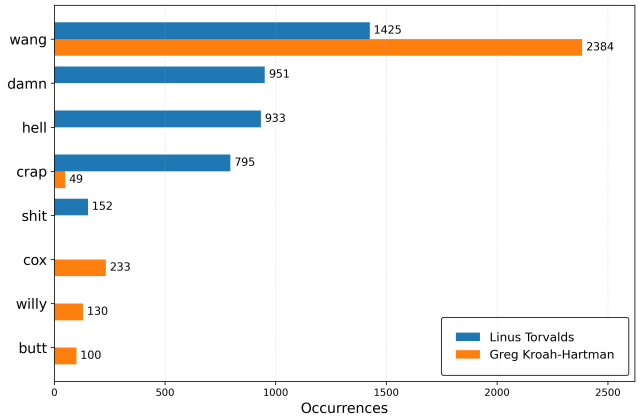

Figure 1: Most frequent expletive terms.

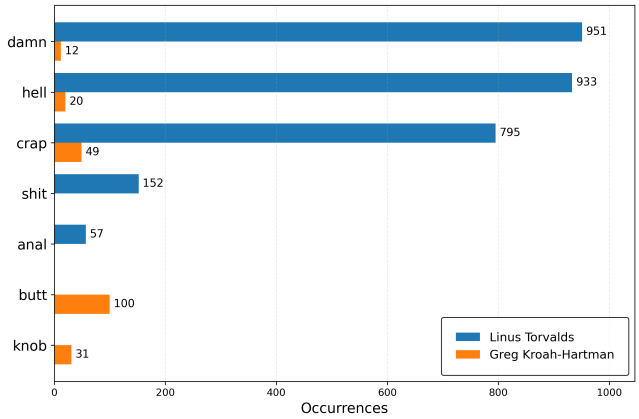

Figure 2: Most frequent expletive terms without ambiguities.

higher rate than Greg Kroah-Hartman, whereas Greg exhibits a considerably higher proportion of singular nouns (NN). Linus, in contrast, shows a relatively higher proportion of plural nouns (NNS). These patterns suggest differences not only in lexical choice, but also in sentence construction and communication style.

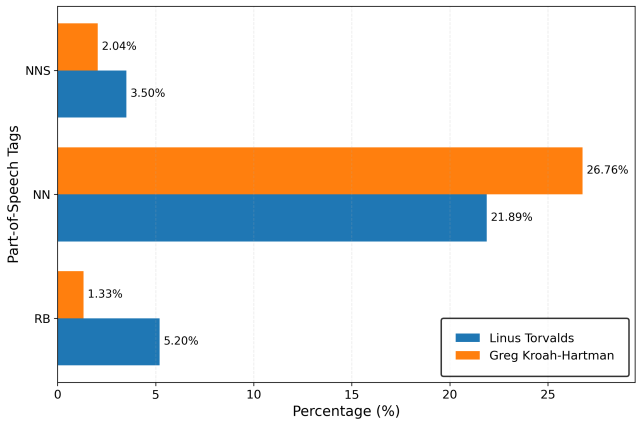

Figure 3: Normalized part-of-speech distribution.

The original study associated Linus Torvalds' higher adverb usage with emphatic and evaluative language, frequently used to reinforce arguments or criticism in technical discussions. In contrast, the greater prevalence of nouns in Greg Kroah-Hartman's messages is consistent with a more implementation-oriented and technical communication style, reflecting discussion topics centered on drivers, subsystems, and software structures.

Overall, the lexical features identified by the classifier further reinforce stylistic differences between the two contributors. Table 6 summarizes the most discriminative terms associated with each contributor according to the Naive Bayes model.

Table 6: Examples of highly discriminative words and their relative association with each contributor.

Word	Linus Ratio	Greg Ratio	Word	Linus Ratio	Greg Ratio
just	0.93	0.07	thing	0.96	0.04
drivers	0.10	0.90	really	0.93	0.07
signed	0.04	0.96	actually	0.94	0.06
struct	0.12	0.88	static	0.09	0.91
think	0.94	0.06	wrote	0.81	0.19

The replicated results reveal a clear distinction between conversational and technical lexical tendencies. Terms more strongly associated with Linus Torvalds include discourse-oriented and emphatic expressions such as *"just"*, *"really"*, *"actually"*, and *"think"*, suggesting a comparatively more argumentative and evaluative communication style. In contrast, Greg Kroah-Hartman is more strongly associated with technical and implementation-oriented vocabulary, including *"drivers"*, *"signed"*, *"struct"*, and *"static"*, reflecting the highly technical nature of many of his interactions.

These findings remain qualitatively aligned with the original study, which reported that Linus Torvalds tends to employ more emphatic and conversational language, whereas Greg Kroah-Hartman adopts a comparatively more technical and procedural communication style. The original study also observed that Greg Kroah-Hartman frequently used explicitly appreciative constructions such as *"thanks a lot for the review, I really appreciate it"* [15], further illustrating stylistic differences beyond purely technical vocabulary.

4.4 Discussion

Table 7 summarizes which findings were reproduced, which remained comparable, and which are not directly comparable due to methodological or corpus differences.

Taken together, the results demonstrate that the analytical pipeline of the original communication styles study can be successfully applied to the updated `LKML5Ws` dataset and continues to produce qualitatively consistent signals across multiple dimensions, including classification performance, lexical statistics, discriminative vocabulary, part-of-speech distributions, and expletive usage. Despite differences in corpus construction and message coverage, the replicated analyses consistently reveal distinguishable communication patterns between Linus Torvalds and Greg Kroah-Hartman.

While this replication intentionally preserved the original analytical pipeline to establish a baseline of reproducibility, our findings highlight the necessity for methodological improvements in future iterations. Specifically, the limitations identified in the lexicon-based expletive detection - such as overlapping contributor names and technical terms - demonstrate that legacy string-matching approaches are insufficient for contemporary mailing list corpora.

Table 7: Replication outcomes relative to the original study

Dimension	Original Study (2016)	This Replication (2026)	Outcome
Classification performance	F1 = 96.3%, Accuracy = 96.2%	F1 = 92.7%, Accuracy = 93.0%	Comparable
Adverb usage (LT > GKH)	Observed as discriminative feature	Observed as discriminative feature	Confirmed
Noun usage (GKH > LT)	Observed	Observed	Confirmed
Expletive asymmetry (LT > GKH)	Strong asymmetry reported	Strong asymmetry reproduced	Confirmed
Discriminative vocabulary patterns	Conversational vs. technical distinctions	Same directional tendency observed	Confirmed
Expletive detection precision	Not explicitly discussed	False positives (e.g., *cox*, *wang*, *willy*)	Deviation
Exact lexical frequencies	Historical corpus (1995–2015)	Contemporary corpus (post-2015)	Not directly comparable
Corpus composition	Original preprocessing and dataset	Different contributors, styles, subsystems	Not directly comparable

Moving forward, this established baseline can be strengthened by integrating context-aware natural language processing techniques, such as *Large Language Models* (LLMs) and modern sentiment analysis frameworks to resolve ambiguities that strict algorithmic replication could not overcome.

These findings suggest that established communication analyses remain applicable to modern LKML datasets, while also emphasizing the importance of carefully inspecting corpus construction choices and inherited analytical assumptions.

5 Threats to Validity

We discuss threats to the validity of the study following the taxonomy proposed by Wohlin et al. [18].

Construct Validity. In our study, authorship attribution relies on mapping multiple email aliases, which may introduce mapping inaccuracies. Furthermore, the replicated pipeline relies on string-based matching for expletive detection, which introduces false positives when offensive vocabulary overlaps with contributor surnames or technical identifiers (see Section 4). Additionally, the original expletive lexicon may not fully capture the evolution of informal language in modern LKML discussions.

Internal Validity. To mitigate the distortion of Greg's larger message volume, we randomly downsampled his messages. While this balanced the classes, different random samples could produce slight variations in classification performance. Additionally, missing timestamp metadata for a subset of messages precluded reliable chronological sequencing, restricting our focus to content-based, aggregate linguistic analyses.

Conclusion Validity. Because our LKML5Ws corpus presents a different distribution of aliases and message counts for the two contributors compared to the original work (see Table 3), direct numerical comparisons are limited. Furthermore, the lack of complete temporal metadata prevents us from isolating the exact 1995–2015 timeframe to calculate the precise dataset overlap with the baseline study. However, our much larger initial data pool and random downsampling strategy ensure that a substantial portion of the analyzed texts represents a different slice of communication data. Consequently, our findings represent a qualitative and independent replication over a distinctly reconfigured sample, rather than a strict re-analysis of the exact original corpus.

External Validity. Because this replication focuses exclusively on two prominent contributors within a single mailing list ecosystem, these results demonstrate methodological viability rather than broadly generalizable conclusions about communication in open-source software. Further studies involving diverse datasets and a broader set of contributors are necessary to extend these findings.

6 Conclusion

This paper investigated the feasibility of revisiting an established communication style analysis of the Linux Kernel Mailing List (LKML) under contemporary data conditions. Rather than confirming exact historical measurements, the study evaluates whether communication-style signals survive substantial changes in mailing-list infrastructure, corpus composition, and project evolution.

We applied the analytical pipeline of the original communication styles study to a new dataset, called LKML5Ws. Our results demonstrate that LKML5Ws produced qualitatively consistent results across multiple dimensions, including authorship classification, descriptive statistics, part-of-speech distributions, and expletive usage. Despite differences in corpus construction and metadata completeness, the experiments revealed clear stylistic differentiation between contributors and strong discriminative linguistic signals, consistent with the qualitative claims of the original study.

Overall, this work establishes a transparent methodological foundation for studying communication in the Linux Kernel Mailing List using contemporary datasets. By explicitly documenting data acquisition challenges, dataset-induced constraints, and their impact on analytical outcomes, it clarifies the conditions under which established communication analyses can be revisited and extended.

Future work will primarily focus on continuously re-executing and extending the study as the LKML5Ws dataset evolves and expands over time. The availability of a new dataset creates opportunities for longitudinal analyses capable of capturing temporal shifts in communication style, contributor behavior, and collaboration dynamics across different development periods. Since this study adopted a random sampling strategy, future work will also investigate the sensitivity of both linguistic statistics and classification outcomes under different sampling configurations. Moreover, future studies will evaluate the use of alternative machine learning approaches and modern language-based techniques, including LLMs.

Finally, we plan to explore connections between communication patterns and technical artifacts, including patches, kernel configurations [5], subsystem dependencies, and architectural evolution. By linking discussion data with implementation-level changes, we aim to better understand how communication dynamics influence implementation decisions, software variability, and the evolution of large-scale open-source systems.

Artifact Availability

All artifacts of this study can be accessed in [2]

Acknowledgments

This research was partially funded by the Brazilian funding agencies CAPES/COFECUB (Finance Code 001, Grant 88881.879016/2023-01 and Ma1036/24), CNPq (Grant 406089/2025-6), FAPESP (Grant 2023/00811-0), and FAPEMIG (Grant APQ-01488-24). We also acknowledge the Behring Foundation[3] and the Brazilian company Stone[4] for their financial support.

[3]https://fundacaobehring.org
[4]https://www.stone.com.br/

References

[1] Linux Kernel Organization [n. d.]. *lore.kernel.org: Linux Kernel Mailing List Archives.* Linux Kernel Organization. Retrieved January 29, 2026 from https://lore.kernel.org

[2] Júlia Azevedo, Theo Canuto, Heraldo Borges, Mathieu Acher, and Juliana Alves Pereira. 2026. A Replication Study of Communication Style Analysis in the Linux Kernel Mailing List – Supplementary Material. https://github.com/aisepucrio/LKML-VEM2026. Accessed on: 10/08/2026.

[3] Lorena A. Barba. 2018. Terminologies for Reproducible Research. arXiv:1802.03311 [cs.DL] https://arxiv.org/abs/1802.03311

[4] Thomas Bock, Niklas Schneider, Angelika Schmid, Sven Apel, and Janet Siegmund. 2025. Understanding the low inter-rater agreement on aggressiveness on the Linux Kernel Mailing List. *Journal of Systems and Software* 222 (2025), 112339.

[5] Heraldo Borges, Juliana Alves Pereira, Djamel Eddine Khelladi, and Mathieu Acher. 2025. Linux Kernel Configurations at Scale: A Dataset for Performance and Evolution Analysis. In *Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering*. 1066–1075.

[6] Carolyn D Egelman, Emerson Murphy-Hill, Elizabeth Kammer, Margaret Morrow Hodges, Collin Green, Ciera Jaspan, and James Lin. 2020. Predicting developers' negative feelings about code review. In *Proceedings of the ACM/IEEE 42nd international conference on software engineering*. 174–185.

[7] Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The "Shut the f**k up" Phenomenon: Characterizing incivility in open source code review discussions. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–35.

[8] Isabella Ferreira, Kate Stewart, Daniel German, and Bram Adams. 2019. A longitudinal study on the maintainers' sentiment of a large scale open source ecosystem. In *2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. IEEE, 17–22.

[9] Daniel M German, Gregorio Robles, Germán Poo-Caamaño, Xin Yang, Hajimu Iida, and Katsuro Inoue. 2018. " Was my contribution fairly reviewed?" a framework to study the perception of fairness in modern code reviews. In *Proceedings of the 40th International Conference on Software Engineering*. 523–534.

[10] Anja Guzzi, Alberto Bacchelli, Michele Lanza, Martin Pinzger, and Arie Van Deursen. 2013. Communication in open source software development mailing lists. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, 277–286.

[11] Rafael Passos, Arthur Pilone, Eduardo Mendes, David Tadokoro, and Paulo Meirelles. 2025. *LKML5Ws: The What, When, Who, Where, and Why in the Linux Kernel Mailing Lists.* doi:10.5281/zenodo.17567225

[12] Rafael Passos, Arthur Pilone, David Tadokoro, and Paulo Meirelles. 2025. DUKS: visualizações e análises unificadas para o Kernel Linux. In *Workshop de Visualização, Evolução e Manutenção de Software (VEM)*. SBC, 90–95.

[13] Huilian Sophie Qiu, Bogdan Vasilescu, Christian Kästner, Carolyn Egelman, Ciera Jaspan, and Emerson Murphy-Hill. 2022. Detecting interpersonal conflict in issues and code review: cross pollinating open-and closed-source approaches. In *Proceedings of the 2022 ACM/IEEE 44th International Conference on Software Engineering: Software Engineering in Society*. 41–55.

[14] Nicolas P Rougier, Konrad Hinsen, Frédéric Alexandre, Thomas Arildsen, Lorena A Barba, Fabien CY Benureau, C Titus Brown, Pierre De Buyl, Ozan Caglayan, Andrew P Davison, et al. 2017. Sustainable computational science: the ReScience initiative. *PeerJ Computer Science* 3 (2017), e142.

[15] Daniel Schneider, Scott Spurlock, and Megan Squire. 2016. Differentiating communication styles of leaders on the linux kernel mailing list. In *Proceedings of the 12th international symposium on open collaboration*. 1–10.

[16] Klaas-Jan Stol and Brian Fitzgerald. 2018. The ABC of software engineering research. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 27, 3 (2018), 1–51.

[17] Xin Tan and Minghui Zhou. 2019. How to communicate when submitting patches: An empirical study of the linux kernel. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–26.

[18] Claes Wohlin, Per Runeson, Martin Höst, Magnus Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in Software Engineering* (1st ed.). Springer Science & Business Media.