Differentiating Communication Styles of Leaders on the Linux Kernel Mailing List

Daniel Schneider
Elon University
Department of Computing Sciences
dschneider2@elon.edu

Scott Spurlock
Elon University
Department of Computing Sciences
sspurlock@elon.edu

Megan Squire
Elon University
Department of Computing Sciences
msquire@elon.edu

ABSTRACT

Much communication between developers of free, libre, and open source software (FLOSS) projects happens on email mailing lists. Geographically and temporally dispersed development teams use email as an asynchronous, centralized, persistently stored institutional memory for sharing code samples, discussing bugs, and making decisions. Email is especially important to large, mature projects, such as the Linux kernel, which has thousands of developers and a multi-layered leadership structure. In this paper, we collect and analyze data to understand the communication patterns in such a community. How do the leaders of the Linux Kernel project write in email? What are the salient features of their writing, and can we discern one leader from another? We find that there are clear written markers for two leaders who have been particularly important to recent discussions of leadership style on the Linux Kernel Mailing List (LKML): Linux Torvalds and Greg Kroah-Hartman. Furthermore, we show that it is straightforward to use a machine learning strategy to automatically differentiate these two leaders based on their writing. Our findings will help researchers understand how this community works, and why there is occasional controversy regarding differences in communication styles on the LKML.

CCS Concepts

• Software and its engineering □ Collaboration in software development • Software and its engineering □ Open source model • Computing Methodologies □ Natural language processing • Computing Methodologies □ Discourse, dialogue, and pragmatics

Keywords

Open source software; Linux; data mining; text mining; email; machine learning; natural language processing.

1. INTRODUCTION

Linux is an operating system initially released by Linus Torvalds in 1991 as an open source project, and is now maintained by Torvalds and a loosely affiliated team of geographically and temporally dispersed software developers. As the software is open source, anyone can read the code or suggest changes to it. However, the people who make the final determination of what changes will be accepted and how those changes should be implemented are Torvalds and a smaller team of trusted "maintainers." These maintainers are in charge of reviewing suggested code fixes, or patches, for the different pieces of the core Linux operating system [1]. To preserve a transparent, archived history of discussions and decisions relating to the project, the entire team - including all developers and interested users - uses email mailing lists to communicate. As of this writing there are 148 different mailing lists associated with the core Linux operating system [2]. The Linux Kernel Mailing List (LKML) is one of those 148 lists. With over 7000 subscribers sending nearly 300 messages per day, it is a very active community. This mailing list has been in use for over 20 years, and includes over 2.16 million emails from approximately 54,000 senders.

Because of its longevity and its size, the LKML is a trove of information about how this very important piece of free, libre, and open source software (FLOSS) is made and how its community works. However, a recent controversy over alleged aggression and intolerance in the LKML suggests additional reasons to take a closer look at this important mailing list. In the July 2013 a dispute erupted on the LKML, centered on allegations surrounding the language and tone that Linus Torvalds has taken towards some other developers in the community. The dispute began when Sarah Sharp, maintainer of the USB 3.0 host controller driver for Linux, accused Torvalds of "advocating for physical intimidation and violence...[and]...advocating for verbal abuse" [3] after Torvalds encouraged another maintainer, Greg Kroah-Hartman, to stop "[acting] as a doormat" and suggested he "may need to learn to shout at people." [4] Another maintainer, Ingo Molnar, encouraged Greg to "be frank with contributors and sometimes swear a bit." [5] As the dispute escalated, Sharp accused Torvalds of being "one of the worst offenders when it comes to verbally abusing people and publicly tearing their emotions apart", but Torvalds was unapologetic, saying "I honestly despise being subtle or "nice".... I simply don't believe in being polite or politically correct." [6]

After a few days of back-and-forth between dozens of community members, the dispute seemed to cool down, with the affected parties agreeing to take up the issue at the Kernel Summit to be held later that year. [7] There is no record on the LKML of whether those discussions took place, but nearly two years after the initial incident, in March 2015, a patch was proposed to the Linux kernel called the Code of Conflict. [8] Its name includes a play on words poking fun of the "Codes of Conduct" found in other FLOSS projects. The LKML Code of Conflict outlined guidelines for appropriate discourse in the LKML, as well as a brief procedure for mediating conflicts between community members. It was written by Kroah-Hartman, the center of the original controversy, and was signed off by 60 other maintainers. Torvalds had made prior negative statements against these kinds of codes. For example, in July of 2013 he said,

OpenSym '16, August 17–19, 2016, Berlin, Germany.
 ISBN 978-1-4503-4451-7/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2957792.2957801

> *...venting of frustrations and anger is actually necessary, and trying to come up with some 'code of conduct' that says that people should be 'respectful' and 'polite' is just so much crap and bullshit.* [9]

Nonetheless, in 2015, this Code of Conflict was ultimately approved for the kernel by Torvalds himself. Later that year, in October 2015, Sharp left the Linux community, stating in part that

> *...[t]op Linux kernel developers often yell at each other in order to correct each other's behavior. That's not a communication style that works for me.* [10]

A month later, in November of 2015, Torvalds again made news headlines for the tone of some comments he made on the LKML to a maintainer. [11]

In the wake of this controversy, we were interested in identifying whether there are in fact differences in discourse patterns of the leaders on the LKML. Specifically, we will use a natural language processing and machine learning strategy to describe and then differentiate the written discourse patterns of Linus Torvalds and Greg Kroah-Hartman, since it was the alleged difference in their communication styles that initially caused the July 2013 dispute. Our research questions are twofold:

RQ1: Considering the two LKML leaders who were at the center of the 2013 controversy, what are the interesting features of their written discourse? Do they have a similar style or is their style very different?

RQ2: Can we automatically differentiate emails written by each person, solely based on their content? What features of the email content are most helpful to this task?

In Section 2 we will discuss related work in email data mining, with particular attention to mining for patterns of power, and mining software development mailing lists. In Section 3 we will outline our methodology and results, including identifying our data sources, explaining our cleaning process, outlining our descriptive statistics, and describing the development of our classification procedure. Next, in Section 4, we will explore the limitations of the study and suggest future research directions before concluding in Section 5.

2. RELATED WORK

In this section, we will review the body of related background literature. Relevant work includes studies revealing patterns of leadership speech or power discourse in electronic workplace communication, and studies that describe how to mine email from software projects, FLOSS projects in particular.

To explore the way group hierarchy is revealed or defined by workplace electronic communications, researchers have traditionally used either a computational linguistics approach or a case study approach. A computational approach tries to discover generalizable techniques for describing dialogue patterns in group electronic communication, for example, differences in vocabulary and grammar [12], formality of communication [13], or usage of certain words and phrases indicative of a leadership role [14, 15, 16]. In [17, 18] Prabhakaran *et al.* study workplace emails for evidence of how individuals with varying levels of power interact, and propose a system for predicting whether a given individual exerts situational power or not. Gilbert [14], trains a system that can learn which phrases signal power in both an upwards and downwards direction using the labeled power hierarchy present in the Enron data set. Other research shows how users express dominance or power in non-email electronic communication, for example on Wikipedia online forums [19, 20].

We use this prior work as inspiration for our research. We know that most FLOSS projects, including the Linux kernel, use email as one of the preferred means of group communication. This is due in part to its ubiquity, its low barriers to entry and platform /technology agnosticism, and the ease with which email can persist through text archives thus preserving a group history [21]. Email archives from FLOSS projects have helped researchers understand who talks on open source mailing lists [22], what they talk about [23], when they talk [24], and how [25]. Our contention is that we can use our email archives to learn about the ways that two of the leaders of the Linux kernel project write to their community, and then subsequently we can use the features of their writing to discern their identities automatically. In the next section we describe our email data set, how we cleaned it, and the various analyses we performed on it.

3. DATA AND METHODS

3.1 Data collection and cleaning

The data collected for this project includes the headers and bodies of nearly 46,000 emails sent to the LKML between June 1995 and February 2015. All the emails considered for this data set were written by either Linus Torvalds or Greg Kroah-Hartman. This dataset was collected from Indiana University's Linux Kernel Mailing List Archive [26].

Because for this project we are only considering emails from two authors, we had to take into account many different email aliases used by each sender over the years to ensure that we only used emails from these two people. Table 1 shows some of the email aliases, as well as the count of how many messages used that alias. In some later messages, the email addresses had been obscured with 'x' characters.

Table 1. Email alias table

Sender	Email	Count
Greg Kroah-Hartman	greg@kroah.com	13,730
Linus Torvalds	torvalds@linux-foundation.org	10,365
Linus Torvalds	torvalds@transmeta.com	5,959
Greg Kroah-Hartman	gregkh@suse.de	5,179
Linus Torvalds	torvalds@osdl.org	4,169
Greg Kroah-Hartman	gregkh@linuxfoundation.org	3,348
Greg Kroah-Hartman	gregkh@xxxxxxxxxxxxxxxxxxx	1,632
Linus Torvalds	torvalds@xxxxxxxxxxxxxxxxxxx	1,139
Greg Kroah-Hartman	greg@xxxxxxxxx	235

HTML versions of the emails, including headers and body text, were downloaded and stored in a MySQL database for processing. The headers include sender name, sender email, timestamp, subject, body text, and the URL where the email came from on the web archive. Email attachments were not collected. To ensure that the body text only includes the intentional written words of our two authors, we cleaned the email bodies by removing HTML

code, source code, reply text, and text that followed a set template or which consisted only of cut-and-pasted text written by others.

To clean the remaining emails, we removed the sender's signature line as well as the automated opening line in a reply email. For example, ***On [date], [author] wrote:*** and ***Message sent to you by [author]*** or the like. We also removed lines that were added by the mailing list software itself (e.g. ***To unsubscribe from this list***, and ***More majordomo info at***, and ***Please read FAQ at***).

We also removed reply text. We observed five different markers for denoting reply text: the presence the <blockquote> tag within the email, the presence of one or more > characters at the beginning of the line, the line beginning with a : (colon) character, the line beginning with |> (pipe character followed by right hand angle bracket), or the line containing <EM>.

We also elected to remove source code from the emails. Our cleaning routine looked for --- and +++ symbols indicating what code has been removed or added in a proposed code snippet. We also added a detector for common closing statements in C code (the language used in Linux kernel). We stripped out any lines in the email that end with }, */, ;, or //. All of these signal either the end of a C code statement or the beginning of a comment. Additionally we remove lines that have /*, which signal the start of a comment, as well as any lines that begin with @@, struct, if(, -#, -}, index, diff, or include/.

Next we removed the numerous HTML tags that had been added by Indiana University for their web site formatting. For example, <p> and
 tags had been added for whitespace, < and > symbols had been replaced by < and >, and so on. We used BeautifulSoup again to remove extra whitespace, HTML tags, and we replaced converted HTML symbols back to their plaintext equivalents.

This data set of cleaned emails in text format and all the source code for this project have been donated to the FLOSSmole project and is available online [27, 71].

3.2 RQ1: Quantifying differences

In this section we quantify some fundamental differences in how Torvalds and Kroah-Hartman write in email in order to answer RQ1: How do these two leaders of the Linux Kernel project write in email? What are the interesting features of their written discourse?

3.2.1 Differences in tenure and message counts

Torvalds has been on the LKML for longer than Kroah-Hartman, which stands to reason since Linux was his project from the beginning. Figure 1 shows the simple counts of emails sent per author over time. The 2015 numbers are not shown on the graph because we only collected January and February of that year.

For the remainder of this study, we do not consider these message counts since that is not a textual feature, but we are providing the information about how many messages were sent over time to give a sense of the scale of this data set and the LKML as a whole.

3.2.2 Lexical differences

To learn about the content of the emails themselves, first, we count the number of emails written by each of the two authors, and we calculate the average word count and sentence count per email. Next we consider lexical diversity, a measure of similarity in the content of a given text. We calculate the lexical diversity of each email as the count of unique words divided by the count of total words, and take the average for each author. A higher lexical diversity score indicates that the author has used more unique words. These results are shown in Table 2.

Figure 1. Message Count, per author, per year

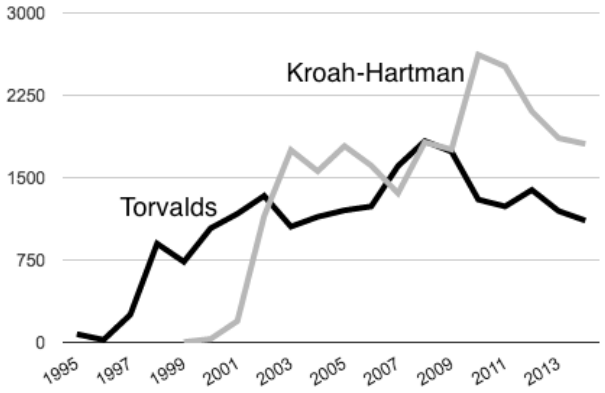

Table 2. Lexical and readability differences

Measure	Torvalds	Kroah-Hartman
Total Emails	21,746	24,145
Avg. Word Count/Email	132	53
Avg. Sentence Count	7.27	3.74
Avg. Lexical Diversity	.08	.27

3.2.3 Readability differences

In Table 3 we show results from calculating some simple readability metrics, including the average Flesch Kincaid Reading Ease (FKRE) score, and the average Flesch Kincaid Grade Level (FKGL) for each author. As an indicator of the readability of a piece of text, Flesch Kincaid is not a perfect indicator (its many flaws are outlined in [28]), but it does attempt to combine the number of sentences, number of words in the sentences, and the number of syllables in those words to distinguish the perceived difficulty of a text. For example, a FKRE score of 90-100 is considered "Very Easy" readability, 60-69 is "Standard" readability, and 0-29 is "Very Confusing" readability. These scores are then sometimes converted to a grade level equivalent. For example, a FKGL score of 7 means that a seventh grade (approximately 12 years old) student in the United States would be able to read the document. We used the Python textstat [29] packages to calculate FKRE and FKGL.

Table 3. Lexical and readability differences

Measure	Torvalds	Kroah-Hartman
Avg. FKRE	73.15	81.74
Avg. FKGL	7.46	5.40

3.2.4 Parts of speech differences

Next we identified frequently used parts of speech (POS) for both authors. We used the Python NLTK package for POS tagging [30]. We tokenized the email as a list of sentences and then tokenized those sentences and classified the words as parts of speech. Table 4 shows the aggregated totals, proportions of usage for each author, and ratios. The final column shows the Torvalds (LT) and Kroah-Hartman (GKH) ratios for each part of speech. The values in Table 4 do not add to 100 for each author since

some parts of speech were not included in the table, for example, conjunctions and interjections.

To assist the readability of Table 4, we have combined a few of the default NLTK POS categories. We combined:

- adverb categories (Adverb, Adverb Comparative, Wh-adverb, Adverb Superlative),
- noun categories (Proper Noun,Singular; Noun, singular or mass; Noun, plural; Proper Noun, Plural),
- pronoun categories (Personal pronoun, Possessive pronoun, Wh-pronoun, Possessive Wh-pronoun),
- verb categories (Verb, 3rd person singular present; Verb, non-3rd person singular present; Verb, base form; Verb, gerund or present participle; Verb, past participle; Verb, past tense), and
- adjective categories (Adjective; Adjective, comparative; Adjective, superlative).

Table 4. Parts of speech differences

POS	LT Count	GKH Count	LT % of all POS	GKH % of all POS	LT: GKH Ratio
Adverb	245,657	85,444	9%	5%	1:0.58
Noun	566,921	450,119	20%	27%	0.75:1
Pronoun	195,521	88,591	7%	5%	1:0.76
Prep.	264,963	121,328	9%	7%	1:0.77
Verb	446,342	217,312	16%	13%	1:0.82
Adjective	176,264	108,695	6%	6%	.97:1

The parts of speech shown in Table 4 are in descending order of how imbalanced the ratio is between the authors. Adverb usage is the most different between Torvalds and Kroah-Hartman. On the other end of the table, adjective use is fairly even between the two. Torvalds' adverb use will emerge as a significant factor in answering RQ2, and we provide more detail in Sections 3.3 and 4 about this.

3.2.5 Expletive usage differences

Next, we hypothesized that the presence of words considered impolite, rude, aggressive, or offensive may also be a distinguishing feature of the way Torvalds and Kroah-Hartman write in email. If we found significant differences in expletive usage, it might help explain why the two could be perceived to have such different tones in their writing.

For our list of expletives, we used the list of words [31] reverse engineered by Jamie Wilkinson from Google's *What do you Love* project [32]. His WDYL expletive list contains words that could be considered either mild ('damn') or strong ('shit') profanity as well as various slang for body parts ('ass'), bodily functions ('piss'), personal insults ('retard'), and the like.

Table 5 shows the counts for emails with the most frequently used expletives, for both leaders. We counted each expletive once, regardless of how many times the same one appeared in an email. For example, Table 5 indicates that Torvalds used the word 'crap' in 1204 different emails. 1204 emails represents about 5.5% of the 21,746 Torvalds emails in the data set.

Table 5. Count of unique expletives used in email

Expletive	Torvalds Expletive Count	Kroah-Hartman Expletive Count
Total Count	*3,090*	*150*
Crap	1204 (5.5%)	107
Hell	725 (3.3%)	22
Damn	682 (3.1%)	2
Shit	126	1
Anal	54	0
Bullshit	50	2
Ass	46	6
God	34	1
Screw	33	0
Bastard	29	0
Bitch	17	0
Piss	17	4
Retard	14	0

There are a few important notes about how this table was constructed.

- Percentages are only shown for expletives that appear in more than 1% of that person's total email count.
- To create this table, related terms have been combined (e.g. ass/-es/arse, crap/-py, fuck/-ed/-ing, bitch/-es/-ing, screw/-ed-/-ing, shit/-ty).
- We added the word 'bullshit' to the original expletive list (it is combined with 'shit' in the table), as it appears 50 times in Torvalds' writing and twice in Kroah-Hartman's.
- We did not search for obscured expletives (e.g., f*ck, sh!t). We have found through reading the emails that Torvalds especially seems to use these a lot, however he changes the format of the obfuscation making them difficult to find and count. We point out three unrelated email examples later in Sections 4.3, 4.5, and 4.6 that all happen to have obscured expletives in them.
- Expletives that may have appeared in source code, attachments, or subject headers were not counted. We only used body text of the emails.
- This expletive list includes English expletives only. This is important because Torvalds has used Finnish expletives such as 'perkele' (and variants) on a few occasions, particularly during the July 2013 dispute with Sharp [6], and in another July 2013 message when he comments,

> *There aren't enough swear-words in the English language, so now I'll have to call you perkeleen vittupää just to express my disgust and frustration with this crap.* [33]

Despite these limitations, there are clear differences in the writing of Torvalds and Kroah-Hartman and, for RQ1, we conclude that they do seem to have some key differences in how they use text, including quantitative measures and in their choice of words.

3.3 RQ2: Naïve Bayes Classifier

Based on our discovery that there are clear differences in the writing styles of these two LKML leaders, we next construct a machine-learning-based classification model to predict whether a given email was written by Torvalds or Kroah-Hartman. This model directly addresses RQ2: Can we automatically differentiate emails written by each person, solely based on their content? What features of the email content are most helpful to this task? Our classification scheme will reveal to what extent we can automatically differentiate between the leaders by their writing alone. Additionally, leveraging statistical analysis of the importance of text-based features for classification, provides further insight into question RQ1.

As a primary tool for our analysis, we employ statistical classification of LKML emails. By first constructing a model for prediction of authorship based on features derived from email text, we can gain insight into the relative importance of these text-based features for classification.

Starting with our cleaned data set of messages sent by Torvalds and Kroah-Hartman, we preprocessed the emails to remove features that would be highly predictive but uninteresting, such as the author's own name appearing in the signature line. From Torvalds emails we removed the words (of upper or lower case) 'linus', 'torvalds', apostrophes, and any sequences of the letter 'x' greater than two. These x's were removed as they were used to signal a break between written language and code, but were not meaningful for classification. The list of words removed from Kroah-Hartman emails includes 'greg', 'kroah', 'hartman', 'kh', apostrophes, and lists of x's greater than two. Note that we did not remove the word 'linus' (or 'torvalds') from the Kroah-Hartman emails, since many times he is talking about Linus to others. (Nor did we remove references to Kroah-Hartman from the Torvalds emails, since we wanted to be able to compare whether Torvalds talked about Kroah-Hartman as much as Kroah-Hartman talked about him.) We also set all words to be lowercase and removed 318 common English stop words, such as 'a', 'an', and 'the', which provide little discriminative power.

Next, taking a bag-of-words approach, we extracted term frequency-inverse document frequency (TF-IDF) features for each email. TF-IDF measures how frequently a term occurs in a document, balanced with how common the term is in the corpus overall. Based on the findings from the POS and expletives analyses described earlier, we also added two additional custom features. Both expletive count and adverb usage proved to be highly predictive for email authorship.

These features serve as input to train a Naïve Bayes Classification (NBC) model. For evaluation, we followed a Monte Carlo cross-validation protocol, randomly splitting the data into training (80%) and validation (20%) sets and averaging results over 10 trials. Table 6 shows the results, averaged over the 10 trials, in terms of accuracy, precision, recall, and F-measure.

Table 6. Classifier results

	Average over 10 trials
Accuracy	96.2%
Precision	97.0%
Recall	95.7%
F1-measure	96.3%

Table 7 shows the results as a confusion matrix, where each row represents the actual class, and each column is the predicted class.

Table 7. Confusion Matrix (counts)

	Predicted: Torvalds	**Predicted: Kroah-Hartman**	**Total**
Actual: Torvalds	4,622.6	206.4	4,829
Actual: Kroah-Hartman	145.3	4,204.7	4,350
Total	4,797.9	4,411.1	9,179

3.3.1 Feature Importance

To identify which features are most helpful to distinguish between the two email authors, we performed a chi-squared test to measure the extent to which each feature correlates with authorship. Table 8 shows the most important features and associated scores (higher score indicates stronger correlation). While these scores help to identify important features, they do not indicate for which author a particular feature is indicative. Thus, for each of these important features, we evaluate the class-conditional probabilities learned by the NBC model. The right-most two columns of Table 8 show the associated normalized probability of predicting each author. How predictive is each feature for determining authorship?

Table 8. Features predicting email authorship

Feature	**Score**	**Prob. Torvalds**	**Prob. Kroah-Hartman**
thanks	1,472.41	2%	98%
adverbCount	416.63	75%	25%
git	303.77	15%	85%
actually	299.94	90%	10%
tree	282.96	13%	87%
applied	264.91	10%	90%
stable	251.57	9%	91%
sorry	243.61	5%	95%
page	237.96	94%	6%
usb	213.76	9%	91%
linux	208.93	14%	86%
thing	204.98	77%	23%
expletivesCount	203.43	94%	6%
just	201.79	66%	44%
org	181.30	8%	92%
sysfs	168.48	3%	97%
kernel	162.86	24%	76%
resend	153.75	1%	99%
Linus	142.54	1%	99%
maybe	141.49	91%	9%

4. DISCUSSION

In this section we discuss specific examples to help explain some of the features shown in Table 8, and how the two authors express these features differently in their writing. Because this paper was motivated by a claim of difference in leadership style and tone, we focus on the features that may explain these style and tone differences, not on the features that are software-focused or task-focused. For instance, in the LKML context, the words 'resend' or 'usb' are more about tasks and less about tone, whereas the word 'sorry' or 'thanks' are more about tone and less about specific jobs or tasks.

4.1 Thanks

In terms of the classification scheme and the features shown in Table 8, the single most discriminative feature between Torvalds and Kroah-Hartman was the latter's use of the word 'thanks'. Many times he uses this as a sign-off (*"**Thanks**, greg k-h"*), although because he often formatted, capitalized, and punctuated this word differently, and because he did not use it in every email, we decided to leave it in the model. Non-signature examples of Kroah-Hartman thanking reviewers and contributors include sentiments like *"**thanks** a lot for the review, I really appreciate it"* [34] and *"**Thanks** to [name] for pointing out my mistake"* [35] and *"**Thanks** to [name] for the big patch, and all of the other people who helped figure this out."* [36]

Torvalds used this same pattern for 'thanks', but it was done far less frequently. Sometimes he uses 'thanks' in a sign-off, and like Kroah-Hartman does, and sometimes he thanks particular people or short lists of people for the work they have done. Still, the difference in how often they each use 'thanks' is the single most discriminating feature in our NBC model.

4.2 Sorry

Kroah-Hartman said 'sorry' enough times for that to be a discriminating feature, also shown in Table 8. Examples of Kroah-Hartman expressing sorrow include nearly 200 instances of 'sorry about that' in various contexts, another 200 'sorry for', another 40 'sorry I' statements, as well as specific apologies like *"**sorry**, my script isn't that smart..."* [37] and *"**sorry** it's late"* [38] and the like. Torvalds also used the word 'sorry' in similar ways, but just not as often as Kroah-Hartman.

Both of them use 'sorry, but' expression to soften a blow. For example, Kroah-Hartman has about 95 instances, such as *"**Sorry, but** you're a bit late"* [39] and Torvalds has about 40 instances such as *"Ugh. **Sorry, but** this patch just looks stupid"* [40].

4.3 Adverbs

Unlike Kroah-Hartman, Torvalds was distinguished by his use of adverbs. Torvalds' most frequently used adverbs include:

- Actually
- Always
- Basically
- Never
- Only
- Really
- Totally
- Very

Sometimes Torvalds uses adverbs to negate something that someone else said, as in *"but that's not **actually** true at all"* [41] or *"Let's hope it **actually** works. Because otherwise this was just a **totally** pointless pain in the *ss."* [42]

Other times he uses adverbs to emphasize a contradiction of an opposing argument, as in

> *The other point of irritation was that there **really** was a lot of stuff that came in yesterday and **basically** treated the merge window as some kind of high-tech limbo dance.* [43]

He also uses them to try to emphasize the problems he finds with an opposing viewpoint:

> *What we do *not* want to allow is to have people call 'readahead' functions and **basically** kill the machine because you now have a unkillable IO that is **insanely** big.* [44]

Many times Torvalds uses multiple adverbs in a single sentence, seemingly for emphasis. For example,

> *Right. And that's **basically** how this 'patch' was actually tested **originally** - by doing this by hand, without **actually** having a patch in hand. I told people: this seems to work **really** well. Mike made it work **automatically**.* [45]

While Kroah-Hartman also uses adverbs, Torvalds uses them much more frequently in his writing.

4.4 Expletives

While no one particular expletive showed up on the list of discriminating Torvalds words in the classifier, the overall usage of expletives did turn out to be an important feature for distinguishing our two authors. Here we describe some of the particulars of the way the expletives are used.

4.4.1 Crap

Of the expletives we looked for, the word 'crap' was a favorite of Torvalds, and to a lesser extent, Kroah-Hartman as well. Prior work has shown that 'crap' is one of the more popular mild profanities on other related FLOSS social media, such as the Ubuntu IRC chat. [46]

The word 'crap' can be used as both a noun (more common) or as an adjective (less common). In the following example, Torvalds uses 'crap' first as an adjective and second as a noun:

> *It's better than horrible **crap** code that doesn't warn, but also doesn't make any *sense*. Because code like this is just **crap**:....*[47]

4.4.2 Shit

The case of 'shit' is similar. Torvalds uses 'shit' as both a noun, and an adjective. In the following exchange he uses crap and shit as synonyms for code, and as multiple parts of speech.

> *Seriously, this whole discussion has been completely moronic. I don't understand why you even bring **shit** like this up: I mean, really? Anybody who writes code like that, or any compiler where that "control_ dependency()" marker makes any difference what-so-ever for code generation should just be retroactively aborted. There is absolutely *zero* reason for that "control_dependency()" **crap**. If you ever find a reason for it, it is either because the compiler is buggy, or because the standard is so **shit** that we should never *ever* use the atomics. Seriously. This thread has devolved into some kind of "just what kind of idiotic compiler cesspool **crap** could we accept". Get away from that f*cking mindset. We don't accept *any* **crap**.* [48]

4.4.3 Hell

Torvalds uses 'hell' frequently as part of a construction with what/why/how/who, as in,

> *I'm upset. How the f*ck did this get into your tree in the first place, and after it got into the tree,* ***WHY THE HELL*** *DID YOU SEND THIS CRAP TO ME?* [49]

He also uses 'hell' to intensify a command such as 'shut up' or 'get out' as in,

> *You have two choices:*
>
> *- acknowledge and fix regressions*
>
> *-* ***get the hell out*** *of kernel development.* [50]

Or,

> *And if they aren't,* ***SHUT THE HELL UP****, because they are total freeloaders, and claimign [sic] that they "support" me is total crap.* [51]

4.4.4 Bitch

The word 'bitch' is used by Torvalds in three ways. First, to mean a difficulty, for example *"Backwards compatibility is a bitch, though"* [52] and *"It can be a real bitch to integrate old filesystems…."* [53]

He also uses the term 'bitch' as a verb to mean whining, as with:

> *And I haven't seen you make CVS usable - I've only seen you bitch, moan, and complain.* [54]

Finally, Torvalds also sometimes uses the term 'bitch' as a slur for a female, presumably to emasculate developers or mock their abilities, as in:

> *Christ guys. This whole thread is retarded. The *ONLY* reason people seem to have for reverting that is a "ooh, my feelings are hurt by how this was done, and now I'm a pissy* ***bitch*** *and I want to get this reverted". Stop the f*cking around already.* [55]

And in the adjective form, he may turn the insult on himself:

> *This is what people get for being difficult to David. I may be a lot more thick-skinned than David, but because of that I'm also a lot more* ***bitchy****.* [56]

4.4.5 Bastard

Torvalds uses 'bastard' in this same self-deprecating way. He turns the insult on himself, describing himself variously as a:

- Callous bastard
- Lazy bastard
- Heartless bastard
- Scheming, conniving bastard
- Paranoid, if careful, bastard
- Bastard and proud of it
- Cheap bastard
- Super-complete bastard
- Opinionated bastard
- Impolite bastard
- Nit-picking bastard
- Absolute bastard

And so on. He does occasionally use the term 'bastard' to refer to someone else, but it is far more often reserved for himself.

4.5 Names

Proper nouns are limited in the word list. Kroah-Hartman refers to Torvalds (using 'Linus') often enough by name that this is a discriminating factor in the classifier, whereas Torvalds refers to Kroah-Hartman much less frequently. This is likely because Kroah-Hartman is a gatekeeper between other developers and Torvalds. When Kroah-Hartman refers to 'Linus', many times he is specifying or clarifying a process. For example,

> *Feel free to play around with this patch, I've sent it on to* ***Linus.*** [57]

Or,

> *Thanks, I've applied this to my trees, and will include it in the next round of changesets to* ***Linus****.* [58]

When Torvalds refers to Kroah-Hartman, it is in a variety of contexts. Sometimes it is to give direction about a the release or testing process, as with,

> ***Greg*** *- I finally got around to merging this tree, and it got [sic] some conflicts"* [59]

Or,

> *"****Greg*** *– please skip these patches from stable for now. I'll try to figure out what's up.* [60]

Other times he is frustrated with some process that Kroah-Hartman is involved in, for instance,

> *And it doesn't help that* ***Greg*** *is sometimes over-eager to take things without them being even in my tree long enough to get much testing.* [61]

Torvalds occasionally chastises Kroah-Hartman or his choice of tools, as in,

> ***Greg*** *seems to use some seriously bad drugs, and creates totally crap commit messages that are just annoying when you have to look at them because there's some conflict.* ***Greg*** *- please fix your crazy tools. Look at this:...and tell me why the f*&% you have commit messages like this….* [62]

Neither of these leaders refers consistently enough to any other Linux contributors for those third party names to become a discriminating feature.

4.6 The thing

The generic noun 'thing' is a Torvalds discriminator (77%). While both authors use 'thing' in their writing, Torvalds uses it much more often, and he uses it in interesting ways. For example, 'the thing' and 'this thing' and 'that thing' are common Torvalds constructions. To compare, Torvalds refers to 'the thing' around 900 times whereas Kroah-Hartman uses this phrase only about 25 times. Torvalds makes reference to 'this thing' or 'that thing' another 200 times, compared to Kroah-Hartman's use of this phrase only around 10 times.

Torvalds' use of the word 'thing' is interesting in other ways besides just in its quantity. Language columnist William Safire explained in a 1986 *New York Times* essay [63] that even though 'thing' is one of the most commonly used words in English, the addition of an attributive noun between 'the' and 'thing', was a new (in 1986) phenomenon. Examples he gave include 'the terrorism thing' and 'the God thing.' Safire supposes that using the word 'thing' with the attributive noun is a way of minimizing the relationship between the speaker and the thing itself. He called this a 'step-back construction,' implying that the speaker is using the phrase 'the [noun] thing' to step away from from the noun being discussed.

The way Torvalds uses 'the [noun] thing' most often seems to be to refer to a particular technical problem and give it a name.

Sometimes he appears to like the thing being discussed and sometimes he does not. Examples:

- *But **an autofs-like thing** might be sufficient* [64]
- *But I don't actuall ysee [sic] **the "hardlink+complex file" thing** as a very hard thing to do necessarily* [64]
- *Note that I'll probably use **the "ALIGN(4096)" thing** for other things too* [65]
- *The obvious breakage is that even though you disabled the IS_SOFT testing for pending signals, you didn't do **the "current-state" thing** right.* [66]
- ***the "task-refcount" thing** is just silly* [67]
- *it was just **this stupid bridge setup thing** that was broken* [68]

However, though minimizing may or may not be the motivator for Torvalds' use of 'the [noun] thing' wording, there are examples where he clearly does use it as a step-back construction. Below is an example where Torvalds uses expletives, step-back construction, and adverbs in a message, along with exclamation points and all capital letters to emphasize his points:

> *Stop the f*cking around already! The whole "we expect ww_ctx to be null" thing shows that YOU DO NOT SEEM TO UNDERSTAND WHAT THE TEST ACTUALLY IS!* [69]

And in a different message,

> *But the "common code helps" thing is WRONG. Face it. It can hurt. A lot. And people shouldn't think it is the God of CS.* [70]

In these cases, Torvalds does appear to use the step-back construction to minimize or trivialize the opposing arguments, not to group them into a convenient named entity.

5. LIMITATIONS AND FUTURE WORK

We acknowledge several limitations to this work, and many avenues for future work. First, as with any project of this size and with this much unstructured and semi-structured data, it is always possible that we may have data cleaning problems. The process for removing source code and reply text from the emails was difficult missed template text or patterned text may have been inadvertently included, unintentionally biasing our NBC model towards one person or the other. A related issue is that it is possible that we may have missed emails due to an odd email header, such as a sender field or email address that was mistyped. While we are confident that the NBC model has enough cases to absorb a few cleaning mistakes here and there, it is always possible that we missed something significant. Since the data set is publicly available [27, 71] we hope that subsequent researchers report any such discrepancies so that a new version can be released.

After completing this project, we have numerous ideas for future work. First, an interesting avenue for investigation would be to study the way the leaders have changed their own writing over time. In this paper we have only considered the 20 years of emails as two giant bags of words, one for each leader. But what differences can we observe in their individual styles over time? Of particular interest would be to collect and compare emails that were sent after Code of Conflict was patched into the kernel with the emails sent before that. We are curious if the lexical characteristics of the emails are different following that action, and if so, in what ways are they different? Initial tests show that there are no significant differences in either Torvalds' or Kroah-Hartman's writing before and after the Code of Conflict was patched, but more time is needed to look closely at these numbers and the new lexical features that may have emerged at that time.

We are also interested in studying some of the other leaders on the LKML besides Torvalds and Kroah-Hartman. Do other leaders on the LKML sound more like Torvalds? What about leaders of other FLOSS projects? Do they have some of the same writing characteristics as Torvalds, or is his style unique?

Within the emails themselves, we identified numerous possibilities for features that might be important, but which we did not account for in this paper. We mentioned obfuscated expletives, and many of the examples shown in Section 4 did include such obscured words. Due to their ubiquity and discriminative power in predicting one author over the other, detecting profanities accurately remains an area of concern. Torvalds in particular uses many different variations on obfuscated profanities, and detecting these should definitely be added into future work.

Another feature that might be fruitful is to attempt to quantify heightened emotional content, aside from the simple presence of profanities. Does one leader tend to rant or "yell" more often? What does ranting or yelling look like? Does yelling get the job done? What happens to an email thread after yelling is introduced? This current paper does not take into account whether there are differences in the authors' use of capitalization or punctuation for emphasis, for example, '_nothing_', '*only*' and 'REALLY' and even the simple exclamation point. We also did not attempt to add sentiment or tone analysis into the NBC model.

6. CONCLUSION

The purpose of this study is to provide data that may answer questions about the perceived differences in communication style, as evidenced in the written correspondence of two of the key leaders of the Linux Kernel project. This issue came to a boiling point in a widely publicized dispute in 2013 in which writing style and tone differences on the LKML were alleged but not detailed. One leader was accused of being too nice, and another leader was accused of being too rude. Since Linux is the largest and most frequently cited success story in FLOSS today, it is critical to understand the inner workings of that project, and whether there any truth to these generalizations about one leader being verbally abusive while another is "too nice".

To help uncover these differences, we first quantified and specified the textual features of the emails sent by Linus Torvalds and Greg Kroah-Hartman to the LKML, the main venue for communication on this project. Next, we used a machine learning strategy to automatically differentiate between emails from these two individuals. Our Naïve Bayes Classification model is highly accurate, when trained on features such as word choice, presence of certain parts of speech, and presence of expletives.

The clear differences we find in the writing styles of these two people, especially in terms of the words they use, do help to validate prior claims by both sides of the debate that indeed these two individuals do use very different patterns of discourse. Examples of these differences include Kroah-Hartman's use of 'thanks' and 'sorry,' and Torvalds' use of expletives, adverbial phrases, and step-back construction. While there are many avenues for future work that can help further explain stylistic differences in email communication on this very important mailing list, our hope is that this paper will be the first step in acknowledging that there are indeed clear, quantifiable differences in discourse patterns between leaders on the LKML.

7. ACKNOWLEDGMENTS

We gratefully acknowledge National Science Foundation (grant number NSF-14-05643) for helping to support this work.

8. REFERENCES

[1] List of maintainers and how to submit kernel changes. Available at https://www.kernel.org/doc/linux/MAINTAINERS

[2] Majordomo lists at vger.kernel.org. Available at http://vger.kernel.org/vger-lists.html

[3] Sharp, S. 2013. Re: [00/19] 3.10.1-stable review. Available at http://lkml.iu.edu/hypermail/linux/kernel/1307.1/03517.html

[4] Torvalds, L. 2013. Re: [00/19] 3.10.1-stable review. Available at http://lkml.iu.edu/hypermail/linux/kernel/1307.1/02653.html

[5] Molnar, I. 2013. Re: [00/19] 3.10.1-stable review. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1307.1/02682.html

[6] Torvalds, L. 2013. Re: [00/19] 3.10.1-stable review. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1307.1/03601.html

[7] Torvalds, L. 2013. Re: [ATTEND] How to act on LKML (was: [00/19] 3.10.1-stable review). Available at: http://lkml.iu.edu/hypermail/linux/kernel/1307.1/03756.html

[8] Torvalds, L. 2015. Merge tag 'cc-4.0-rc2' of git://git.kernel.org/pub/ scm/linux/kernel/git/gregkh/char-misc. Available at: https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=b0bc65729070b9cbdbb53ff042984a3c545a0e34

[9] McMillan, R. 2013. Linus Torvalds defends his god- given right to offend you. *Wired*. July 16. Available at: http://www.wired.com/2013/07/linus-torvalds-right-to-offend/

[10] Sharp, S. 2015. Closing a door. October 2. Available at: http://sarah.thesharps.us/2015/10/05/closing-a-door/

[11] Sharwood, S. 2015. Linus Torvalds fires off angry 'compiler-masturbation' rant . Available at: http://www.theregister.co.uk/2015/11/01/linus_torvalds_fires_off_angry_compilermasturbation_rant/

[12] Bramsen P., Escobar-Molano, M., Patel, A., and Alonso R. 2011. Extracting social power relationships from natural language. *The Association for Computer Linguistics*, 773–782.

[13] Peterson, K., Hohensee, M., and Xia, F. 2011. Email formality in the workplace: A case study on the Enron corpus. *Proceedings of the Workshop on Language in Social Media*, 86-95.

[14] Gilbert, E. 2012. Phrases that signal workplace hierarchy. *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 1037–1046.

[15] Prabhakaran, V. 2012. Predicting overt display of power in written dialogs. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 518-522.

[16] Strzalkowski, T., Broadwell, G.A., Stromer-Galley, J., Shaikh, S., Taylor, S., and Webb, N. 2010. Modeling socio-cultural phenomena in discourse. *Proceedings of the 23rd International Conference on COLING*.

[17] Prabhakaran, V. 2012. Detecting Power Relations from Written Dialog. *In Proceedings of Student Research Workshop, ACL 2012,* 7-12.

[18] Prabhakaran, V., Rambow, O., and Diab, M. 2012. Who's (Really) the Boss? Perception of Situational Power in Written Interactions. *Proceedings of the 24th International Conference on COLING.*

[19] Burke M., and Kraut R. 2008. Mopping up: Modeling Wikipedia promotion decisions. *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW)*, 27-36.

[20] Thom-Santelli, J., Cosley, D., Gay, G. (2009). What's mine is mine: territoriality in collaborative authoring. *Proceedings of ACM Conference on Computer Human Interaction (CHI),* 1481-1484.

[21] Squire, M. 2012. How the FLOSS research community uses email archives. *International Journal of Open Source Software and Processes*, 4(1). 37-59.

[22] Bird, C., Pattison, D., D'Souza, R., Filkov, V., and Devanbu, P. 2008. Latent social structure in open source projects. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, 24-35.

[23] Guzzi, A., Bacchelli, A., Lanza, M., Pinzger, M., and Deursen, A.V. 2013. Communication in open source software development mailing lists. In *Proceedings of the 10th Working Conference on Mining Software Repositories*. 277-286.

[24] Ibrahim, Walid M., Bettenburg, N., Shihab, E., Adams, B., and Hassan, A. E. 2010. "Should I contribute to this discussion?." In Proc. 7th IEEE Working Conference on Mining Software Repositories (MSR).181-190.

[25] Rigby, P., and Hassan, A. 2007. What can oss mailing lists tell us? a preliminary psychometric text analysis of the apache developer mailing list. In *Proc 4th International Workshop on Mining Software Repositories*. 23-30.

[26] The Linux Kernel Archive. Available at: http://lkml.iu.edu/hypermail/linux/kernel/

[27] Data for LKML speech study. Available at: http://flossdata.syr.edu/data/lkml/lkmlspeech2016

[28] Kemper, S. 1983. Measuring the inference load of a text. *Journal of Educational Psychology*, 75(3): 391-401.

[29] Textstat 0.2. Available at: https://pypi.python.org/pypi/textstat/

[30] Categorizing and tagging words. *Natural Language Toolkit Documentation*. Available at: http://www.nltk.org/book/ch05.html

[31] Wilkinson, J. 2010. Google_twunter_lol. Available at: https://gist.github.com/jamiew/1112488

[32] What do you love? Available at: http://www.wdyl.com/

[33] Torvalds, L. 2013. Re: [GIT pull] x86 updates for 3.11 . Available at: http://lkml.iu.edu/hypermail/linux/kernel/1307.1/03003.html

[34] Kroah-Hartman, G. 2008. Re: sysfs: tagged directories not merged completely yet. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0810.0/1517.html

[35] Kroah-Hartman, G. 2007. [GIT PATCH] driver core fix for previous 2.6.24-rc1 pull. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0710.3/3832.html

[36] Kroah-Hartman, G. 2003. [BK PATCH] PCI fixes for 2.6.0-test5. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0309.1/1398.html

[37] Kroah-Hartman, G. 2005. no more "applied, thanks" emails from me.. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0505.0/0616.html

[38] Kroah-Hartman, G. 2009. Staging tree status for the .33 kernel merge. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0912.2/02965.html

[39] Kroah-Hartman, G. 2004. Re: [ANNOUNCE] udev 021 release. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0403.0/0756.html

[40] Torvalds, L. 2013. Re: [PATCH] init: Do not warn on non-zero initcall return. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1305.0/00215.html

[41] Torvalds, L. 2014. Re: Interactivity regression since v3.11 in mm/vmscan.c. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1406.0/05019.html

[42] Torvalds, L. 2012. Linux 3.7-rc1. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1210.1/03687.html

[43] Torvalds, L. 2011. Linux 3.2-rc1. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1111.0/02918.html

[44] Torvalds, L. 2013. Re: [PATCH RFC] mm readahead: Fix the readahead fail in case of emptynuma node. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1312.1/04498.html

[45] Torvalds, L. 2010. Re: [RFC/RFT PATCH v3] sched: automated per tty task groups. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1011.2/00352.html

[46] Squire, M. and Gazda, R. 2015. FLOSS as a source for profanity and insults: Collecting the data. *Proceedings of the 48th Hawaii International Conference on System Sciences.* 5290-5298.

[47] Torvalds, L. 2015. Re: [GIT PULL] aio: fix sleeping while TASK_INTERRUPTIBLE. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1502.0/00135.html

[48] Torvalds, L. 2014. Re: [RFC][PATCH 0/5] arch: atomic rework. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1402.1/05057.html

[49] Torvalds, L. 2013. Re: [git pull] GPIO for v3.10. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1305.0/01484.html

[50] Torvalds, L. 2015. Re: [PATCH] Revert 9fc2105aeaaf56b0cf75296a84702d0f9e64437b to fix pyaudio (and probably more). Available at: http://lkml.iu.edu/hypermail/linux/kernel/1501.0/00786.html

[51] Torvalds, L. 2005. Re: userspace breakage. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0512.3/0986.html

[52] Torvalds, L. 1999. Re: Ingo's PIII FXSTOR patch [2 Mar] causes problems. Available at: http://lkml.iu.edu/hypermail/linux/kernel/9904.2/0081.html

[53] Torvalds, L. 1999. Re: vm kills processes in our 2.3.12 port of reiserfs - what was. Available at: http://lkml.iu.edu/hypermail/linux/kernel/9908.3/1260.html

[54] Torvalds, L. 2003. Re: log-buf-len dynamic. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0309.2/1534.html

[55] Torvalds, L. 2012. Re: [PATCH, 3.7-rc7, RESEND] fs: revert commit bbdd6808 to fallocate UAPI. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1212.0/01745.html

[56] Torvalds, L. 1997. buffer cache patches - anybody willing to summarize?. Available at: http://lkml.iu.edu/hypermail/linux/kernel/9708.0/0130.html

[57] Kroah-Hartman, G. 2002. [PATCH] driverfs support for USB. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0202.0/1592.html

[58] Kroah-Hartman, G. 2002. Re: [PATCH] USB set-bit takes a long tweaks. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0204.2/0136.html

[59] Torvalds, L. 2008. Re: [GIT PATCH] STAGING patches for 2.6.28. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0810.2/1039.html

[60] Torvalds, L. 2010. Re: [stable] [PATCH] Fix 'flush_old_exec()/setup_new_exec()' split. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1002.0/01866.html

[61] Torvalds, L. 2010. Re: stable cc's in linux -next was Re: [BUG] x86: bootmem broken onSGI UV. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1010.1/00626.html

[62] Torvalds, L. 2011. Re: [GIT] Networking. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1110.3/00634.html

[63] Safire, W. 1986. On language: The word's the thing. *New York Times.* June 8. Available at: http://www.nytimes.com/1986/06/08/magazine/on-language-the-word-s-the-thing.html

[64] Torvalds, L. 2000. Re: NTFS-like streams?. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0008.1/1010.html

[65] Torvalds, L. 1997. Re: i386 memory corrupted by free_initmem(). Available at: http://lkml.iu.edu/hypermail/linux/kernel/9705.2/0079.html

[66] Torvalds, L. 1998. Re: PATCH: NFS oops in 2.1.117. Available at: http://lkml.iu.edu/hypermail/linux/kernel/9808.2/1002.html

[67] Torvalds, L. 1999. Re: [patch] zoned-2.3.28-K4. Available at: http://lkml.iu.edu/hypermail/linux/kernel/9911.2/0382.html

[68] Torvalds, L. 2007. Re: PCI resource problems caused by improper address rounding. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0712.2/1137.html

[69] Torvalds, L. 2013. Re: [GIT PULL] locking fix. Available at: http://lkml.iu.edu/hypermail/linux/kernel/1310.3/01202.htm

[70] Torvalds, L. 2000. Re: [PATCH] Re: Move of input drivers, some word needed from you. Available at: http://lkml.iu.edu/hypermail/linux/kernel/0008.2/0188.html

[71] Code for 2016 LKML Speech Study. Available at https://github.com/FLOSSmole/LKMLspeech