What Are We Talking About?

An Analysis of the SIGCSE-Members Listserv

Austin Cory Bart, Clifford A. Shaffer
Virginia Tech
Blacksburg, VA
acbart@vt.edu,shaffer@vt.edu

ABSTRACT

The SIGCSE-Members listserv has been archiving posts by the Computer Science Education community for the past 21 years. This paper characterizes the archive of posts, in order to better understand the nature of the community from a quantitative perspective. We apply a number of email mining techniques, including a topical analysis through N-grams. Threads, posters, and posts are characterized in terms of duration and temporally. We also demonstrate how emails from the listserv can be successfully classified using machine learning algorithms, and report on an unsuccessful attempt to predict thread popularity. All of the scripts we used to collect, process, and analyze the data are freely available in the hopes that other researchers will replicate, refine, and extend our results.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference format:
Austin Cory Bart, Clifford A. Shaffer. 1997. What Are We Talking About?. In *Proceedings of ACM Woodstock conference, El Paso, Texas USA, July 1997 (WOODSTOCK'97),* 6 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

About 21 years ago, the SIGCSE-Members listserv began archiving posts by the Computer Science Education community. This paper characterizes the archive of posts, in order to better understand the nature of the community. To our knowledge, this is the first attempt at a formal, quantitative analysis of the SIGCSE-Members mailing list. Previously, Kim Bruce published a summary of a particularly fervent conversation from the mailing list in an ITiCSE working group [4], which was followed up by a phenomenographical analysis [2]. Otherwise, there has been little reference to the mailing list in formal CS Ed research.

General analyses of email lists and on-line communities has been a wide area of study for decades, and the techniques used can vary substantially. [1] gives a useful overview of various problems and methods by reviewing a plethora of relevant papers in this area. As a subfield of text categorization, typical subproblems include N-gram analysis, temporal information analysis, and thread clustering. We tackle a number of these problems. For this paper, we ignore the most typical form of email classification (spam analysis), due to the highly-curated nature of the SIGCSE-members listserv.

1.1 Audience and Contributions

The primary audience of this paper are members of the Computing Education community who are interested in better understanding this professional community of practice. This paper could also be of broader interest to researchers in more general education who want to observe an exemplar sub-community, or to social interaction researchers interested in a professional education community. Educators that are new to the SIGCSE-Members listserv, and are interested in getting involved in the mailing list, should also find the analysis a useful introduction.

This paper makes the following contributions in service of these audiences:

- The collection, preparation, and release of the email archive into a public repository,
- A time-oriented review of popular post topics through N-gram analysis of expected and common keywords,
- Visualizations of post characteristics based on analysis of thread length, thread duration, and post size,
- Visualizations of temporal posting trends by analyzing posts' submission date and time,
- A description of the frequency of job and conference ads using a supervised classification algorithm,
- A description of poster behavior based on analysis of the number of submissions from each poster, and
- An attempt at predicting thread popularity from quantitative factors.

2 DATA COLLECTION

The SIGCSE-Members listserv is restricted to posting by confirmed SIGCSE members; before posting, users must confirm their ACM membership ID with the listserv administrators via email [8]. Fortunately, the site also maintains a public archive of all posts [1]. This public archive was scraped, processed, and analyzed using a combination of Python scripts. To encourage replication and extension of our analyses, we make these scripts publicly available through a public Git repository [2].

[1] https://listserv.acm.org/SCRIPTS/WA-ACMLPX.CGI?A0=SIGCSE-MEMBERS
[2] https://goo.gl/9MSmA7

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

The following data was collected from the archive about every email posted to the archive from October 1996 to April 2017:

- The body of the email (either as unicode text or HTML).
- The subject line of the email.
- Any attachments to the email, including the MIME filetype.
- The name of the sender (but not their email, as described below).
- The timestamp of the email, with second-level resolution.
- The conversation thread that the email belongs to (and its position within the thread).

Email address normalization is a challenging problem in email mining, because of how names and addresses can be inconsistently aliased between accounts and mail management systems [3]. The SIGCSE archive does not reveal posters' emails addresses unless the viewer logs in as a confirmed member. For simplicity, we only collected data that was publicly viewable without logging in. This means that, although we collected user-friendly names of users, we did not collect the exact email address that sent each email. Therefore, the following methodology was used to normalize posters:

(1) Names were converted to lower case.
(2) Unnecessary titles were stripped out (e.g., "dr" or "doctor").
(3) Each full name was split into a set of names and sorted (e.g., the name "charles babbage" would be converted to the list "babbage, charles") in order to avoid common reorderings of names.

Once the email bodies and subjects were downloaded, the raw data was processed by applying the following transformations:

- Special unicode characters were converted to similar ascii characters, in order to simplify textual analysis. This may have effects on the analysis with regards to international participants in the listserv, suggesting a direction for follow-up studies.
- HTML files were converted to a plain-text representation using the Python Html2Text library [3].
- Quoted text that was repeated in replies were removed.
- Signatures and authorship lines were removed.

Overall, 16526 emails were downloaded and processed using the above methodology. The remainder of this paper is dedicated to describing our own preliminary analyses of this data.

3 TOPICAL ANALYSIS

Figure 1 shows the top 40 longest threads, all of which had at least 20 posts, organized by the year that the thread was made. This list demonstrates the range of topics covered on the Listserv, including debates about best practices ("Value of Unit Testing"), programming language arguments ("Java vs. C"), pedagogical conversations ("Students who get help from online forums"), community and professional conversations ("Faculty who are poor teachers - why do we tolerate them"), and even some humor ("Fun - how do you know if you are an old CS prof").

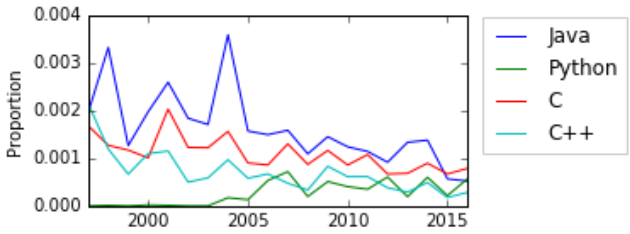
Figure 2 shows the results of targeted unigram and bigram analysis of common phrases over each year. Each plot represents a

[3]https://github.com/aaronsw/html2text

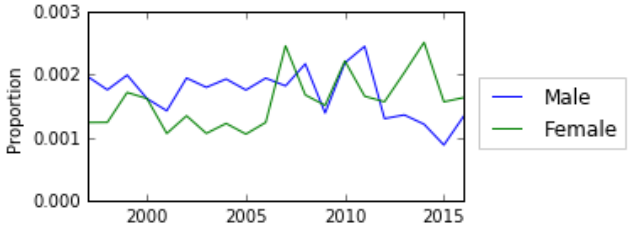
year	subject
1996	More on CSAB
1999	Story on Computer education in todays Chronicle Online
	Grad School
2000	Simplified Java IO for CS1
2004	ACM Java Task Force announcement
	Java vs C
2005	Elimination of Computer Science Courses from the High Schools List of NCAAApproved Core Courses
	Differential Equations
	Decline in CS enrollment was Re Two replie to Duben and to Shaffer
2006	I Object
	Intro to programming course without reference to any languages
	Coder Jobs Painfully Stable
	What price publications
	Java Programming IDE
2007	Carrots Sticks and women
2008	Are CSI and CSII language independent
	CS science or engineering was calculus
2009	Is Obtaining ABET Accreditation for a Doctoral Granting Institutions Undergraduate Program Worth the Effort
	looking for information on how Javafirst schools handle the topic of pointers
	Need for a CS version of the Order of the Engineer
2010	outrage Albion College board axes CS ignoring shared governance
	article in todays Chronicle
	CSI functions first
2011	SIGCSE Robot Hoedown attracts Senators attention
	Designing a CS classroom
2012	Python better than Java for CS1
	Students who get help from online forums
	Resell your books
	Recursion Question
	computer science majors study less on average than elementary education majors
2013	Handwriting code on exams
2014	Summer reading
	Computer Engineering Barbie
	Getting college credit from MOOCs
	systems that detect plagiarism in programming assignments
2016	fun how do you know if you are an old CS prof
	Faculty who are poor teachers why do we tolerate them
2017	workflow for easilly adding feedback to code while grading
	Can we talk about documenting our code
	Value of Unit Testing

Figure 1: The 40 Longest Threads, Ordered by Time

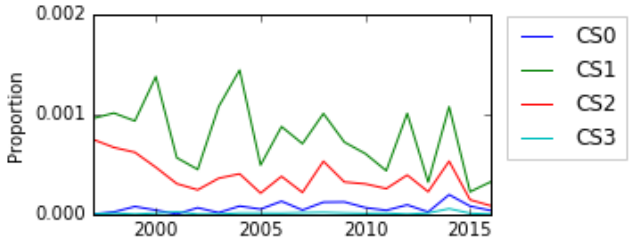
different grouping of phrases and words. The number of occurrences of the word or phrase were divided by the total number of words from that year in order to provide a time-adjusted proportion.



(a) Programming Languages



(b) Gendered Words



(c) CS Levels

Figure 2: N-gram Analysis over Years

Figure 2a shows the word usage rate over years across various programming languages. Java has almost always been the top language discussed. C and C++ are also frequent topics, although somewhat less so. Python has grown dramatically in popularity since the early 2000s. In general, however, the discussion of all of these languages seems to have decreased in the past decade. There was surprisingly little conversation about Scheme, Racket, or JavaScript, to the point where they could not be reasonably included in the visualization.

Figure 2b shows the trends in the use of gendered language over years. Two kinds of words were counted as gendered: "Male" and "Female". Male words include "men", "boys", "gentlemen", "his", "he"; female words include "women", "girls", "ladies", "her", and "she". In the first decade of the mailing list, most years had more male words used than female. However, that trend changed around 2007, which saw the beginning of a more commingled period. In the past half-decade, there has been more female gendered words used than male. We suggest this trend reflects the increased emphasis within the SIGCSE community on the gender balance problem.

Figure 2c shows usage of the terms CS0, CS1, CS2, and CS3 over time. These terms describe introductory courses at various levels [5]. CS1 was easily the most popular term, followed by CS2. CS0 was consistently ranked third, while CS3 was barely ever used (a grand total of 22 times over the two decades).

A few additional visualizations are not included in this paper for space limitations. The term "undergraduate" consistently appeared half as frequently as "graduate". The terms "teaching" and "research" appeared equally often. The terms "K-12", "Middle School", and "High School" were almost unused until 2007, when they began skyrocketing in popularity.

4 POST AND THREAD CHARACTERISTICS

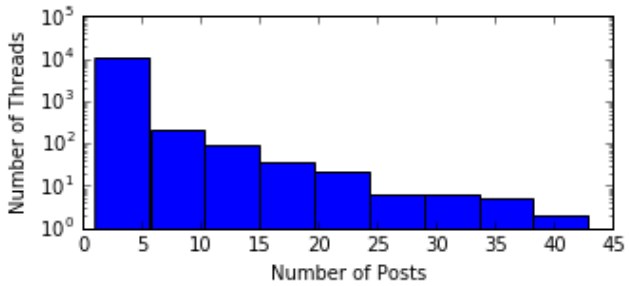


Figure 3: Distribution of Thread Sizes

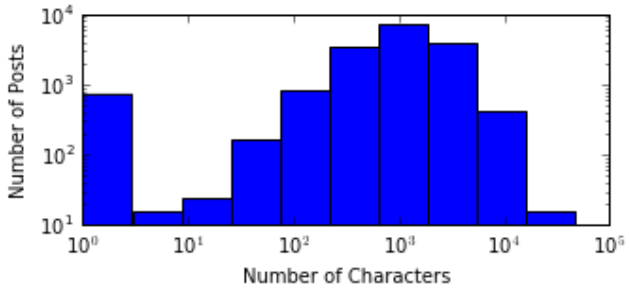


Figure 4: Distribution of Post Lengths

Figure 3 and 4 shows the distribution of email thread sizes and individual email length, respectively. The first graph is logarithmic on the y-axis, while the second is logarithmic on both axes. Most threads tend to be very small, lasting only one or two posts (57.5%). Only one thread was longer than 60 posts, totaling 91 posts ("fun -> how do you know if you are an old CS prof?") and four separate follow-up threads. Similar to thread length, most posts tended to be very short, lasting less than 5000 characters. The overall distribution of posts is log-normal, centered roughly around 1500 characters or about half a page of text. Only one post was more than 40,000 characters long, a particularly long notification about the 1997 IEEE Conference on Software Engineering Education and Training. A large number of posts were less than 10 characters, some of which

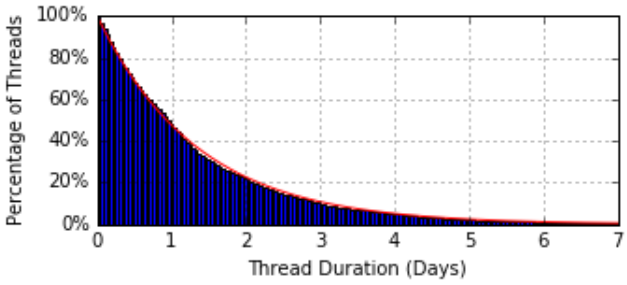


Figure 5: Non-trivial Thread Duration

may be caused by inadequacies of the processing methodology used.

Figure 5 is a bar chart of the duration of non-trivial threads; each bar represents the percentage of threads that last at least that number of days. In this case, non-trivial threads were any thread that had at least one reply [6]. These durations follow a power law, as shown by the red line regression line with the following formula:

$$\%\ of\ Threads\ Ended = \frac{1}{2.1^{days}}$$

The Mean Squared Error of this regression rounds to 0. The implication is fairly straightforward: almost half (47.6%) of all non-trivial threads end after a day, roughly a quarter (22.7%) end after two days, a tenth (10.8%) after three days, and so on. Previous studies of forum and email thread lengths suggest that power-law distributions are fairly common in online communications [3].

5 TEMPORAL ANALYSIS

Figure 6 shows trends over time in user posting behavior. Each of the four graphs show a different granularity of time: over the entire time period, grouped by month, grouped by day of the week, and grouped by the time of day. These posting habits reveal human behavior within the SIGCSE-members listserv. Although not particularly surprising, participants tend to avoid posting during non-work hours, on weekends, and during holiday seasons. These patterns should be considered when waiting on responses or when trying to decide when to post messages.

The Yearly trend shows a chaotic growth over the past two decades. The variation over time is considerably higher than in other graphs. However, linear regression reveals a significant ($p < .01$) but tiny ($slope = .13$) positive trend. This suggests that the rate of posting has relatively increased over the years, but with strong fluctuation – most likely due to the cyclical boom/bust cycles seen in computer science programs. The most sustained activity seems to have occurred during the 2009-2010 time period, while the lowest was during the 2001-2003 time period.

The Monthly distribution reveals that users are disengaged during the summer months and during the winter holidays. Although January quickly peaks with new posts, the rest of the spring is a slow downward trend until posts fall off completely in June. When most US Fall semesters begin in September, posts also pick-up and eventually peak in October, before falling back down in November. It is unclear exactly why posts peak so heavily in January, but two hypotheses are: 1) anticipation of the SIGCSE conference, or 2) to make up for lost time during December. Of course, there is high variation within most months, so these conclusions are tenuous.

The day of week graph shows the frequency of emails sent during days of the week. This graph indicates that for any given day of the week, there are usually no posts. When posts do occur, they tend to happen considerably more often on weekdays than on weekends. This suggests that most posters disengage during the weekend. There appears to be a generally negative trend during the week, at least in terms of the heavily active days. If accurate, then this implies that posters steadily disengage over the course of the week.

The final graph shows the frequency of posts over hours of the day. Similar to the day of week graph, most hours have no emails posted. However, the graph does seem to show three general time periods: a busy morning period, a moderately busy evening period, and a quiet night period. One conclusion is that SIGCSE posters tend to catch up on their correspondence early in the day and trail off over time, mostly disengaging after work. Timezone data was not always available for each post. When it was available, all times were adjusted to Greenwich Mean Time in order to meaningfully compare post times from a common vantage point. Therefore, the 24-hour graph should be taken as an approximation of the actual hourly posting behavior.

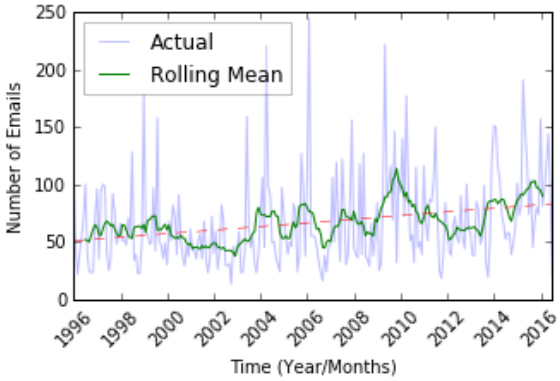
6 JOBS AND CONFERENCES

When reviewing the posts in the archive, a large number of job and conference ads were noted. Machine Learning was used to classify each post based on the type of post, a common technique in email classification [9]. For training data, 303 posts were manually tagged as either "Normal Conversation" (200), "Job Ad" (52), or "Conference Discussion" (51). These posts were then analyzed in a Support-Vector Machine with stochastic gradient descent (SGD) learning, using the body of the post as a Bag-of-Words, via the Scikit-Learn Python package [7]. The performance of the classifier was very impressive, with perfect precision, recall, and F1-scores based on the test data.

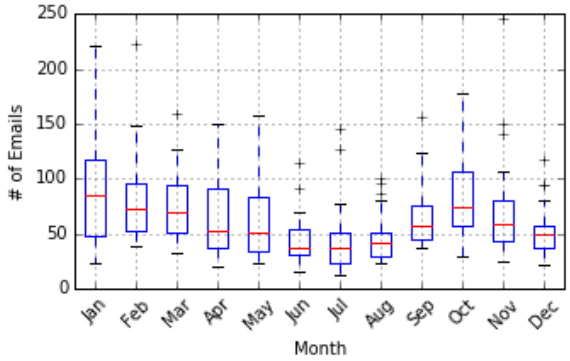
The classifier suggests that 2322 job ads and 2368 conference-related posts have been made on the listserv. The percentage of job ads and conference posts were surprisingly similar (14.3% and 14.0% respectively) over the entire dataset. However, the yearly rate shown in Figure 7 reveals that neither rate has been constant; conference posts have relatively steadily increased slowly, while job ads faced a period of decline during the mid-2000s before resuming growth in the past half-decade. This closely matches the boom/bust cycles previously seen in computing education. Figure 8 shows the monthly trend in job posts. Job ads appear to be seasonal: most posts come out during the fall, and few posts come during the summer (a significant difference, according to a one-way ANOVA with $p < 0.05$). This matches the authors' intuitions about hiring schedules.

7 POSTER BEHAVIOR

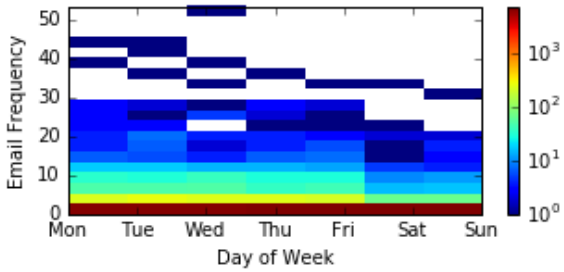
Our next form of analysis was to determine the behavioral patterns of the typical posters to the listserv. Previously, we described how poster names were normalized to more accurately group posts by repeated users. After this process was applied, 2205 estimated



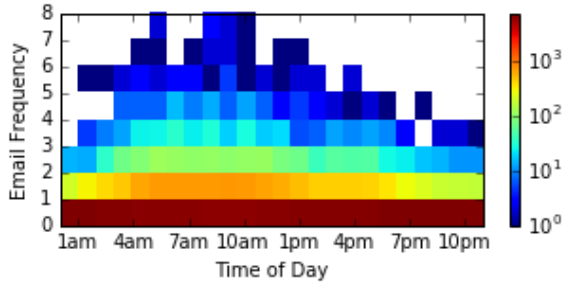
(a) Number of Emails per Year/Month



(b) Distribution of Emails per Month



(c) Frequency Distribution of Emails by Hour



(d) Frequency Distribution of Emails by Day of Week

Figure 6: Posts over Time

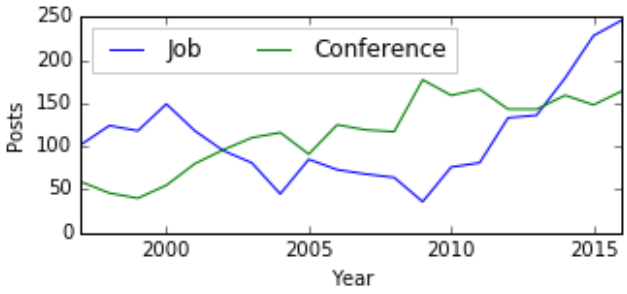


Figure 7: Post Types by year

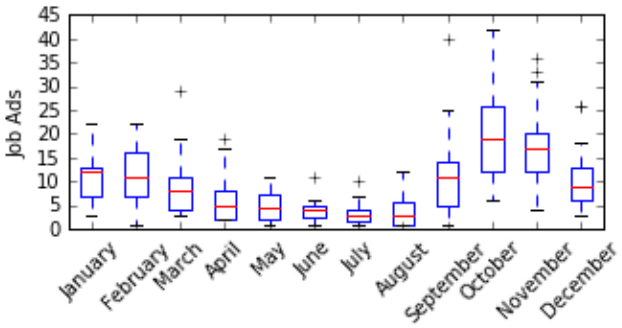


Figure 8: Distribution of Job Ads per Month

unique posters were found. The distribution of posts per poster is given in Figure 9 . The median number of messages per poster is 2, with 51.3% of all posters having 2 or fewer posts associated with their normalized name. Table 1 breaks down the frequency and percentages of four different kinds of users: all users, regular users (>0 posts), active (>20 posts), and super active (>50 posts). From this, it can be seen that the bulk of the posts are made by regular posters, and that active and superactive posters both contribute large numbers of posts, despite making up a relatively small percentage of the total number of posters.

User Type	Threshold	Posters	Posts
All	>0 posts	2205 (100%)	16525 (100%)
Regular	>2 posts	1073 (48.7%)	15102 (91.4%)
Active	>20 posts	179 (8.1%)	8615 (52.1%)
Super	>50 posts	40 (1.8%)	4248 (25.7%)

Table 1: Poster Type Frequencies

All of the active members of the mailing list was identified and manually researched in order to correlate their mailing list activity with their research productivity, as defined by their publication

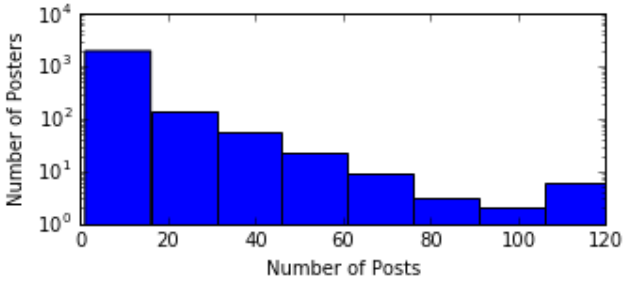


Figure 9: Distribution of Posts per User

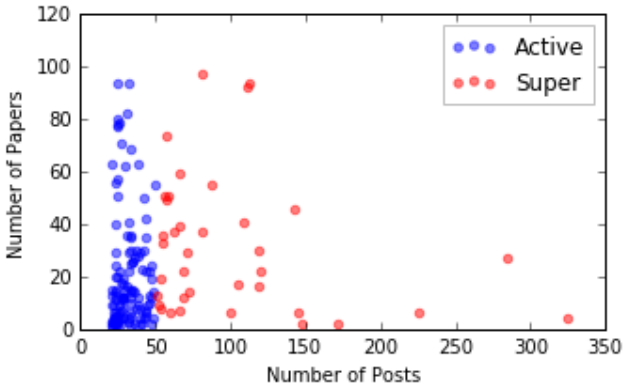


Figure 10: Productivity vs. Activity of Active Users

count within the SIGCSE Collection of the ACM Digital Library [4]. Note that this collection includes several publications, including SIGCSE, ICER, ITiCSE, and Koli. Figure 10 is a scatter plot showing the relationship between the users' posts and their SIGCSE publications for active and superactive users. Surprisingly, there was no significant correlation between activity and productivity. There were a number of users who published frequently without posting to the mailing list, and many active users with few publications to their name within the SIGCSE publication archives.

8 THREAD POPULARITY REGRESSION

For our final analysis, an attempt was made to statistically predict what makes a popular thread. We hypothesized that some subset of the posts' temporal information (i.e. month, hour of day, day of week), post length (characters), and the presence of questions (approximated by the number of question marks), would allow us to predict the size of a thread (which would serve as the thread's popularity). First, we attempted to stepwise fit a linear regression model for each possible subset of the factors. Second, we classified each continuous variable into finite categories (e.g., hour could be classified as "morning", "evening", etc.) and used in a logistic regression model. However, in both cases, no models could be found to reasonably fit the data. In fact, the R^2 was $< .01$ in all such models that we attempted. This suggests that, despite clear patterns in user behavior, there are no predictable ways to increase user response using these simple statistical models generated from these factors.

[4]http://dl.acm.org/sig.cfm?id=SP927

9 FUTURE WORK

At this point, we have run a number of exploratory analyses on the collected archive of posts. However, we believe that there are still many more possible research questions to be explored using this dataset. As previously mentioned, we have open-sourced our collection and processing scripts in the hopes that interested readers will replicate, refine, and extend our investigations. As we review our work, we find a few immediately visible areas for improvement.

As described in our methodology, exact email addresses were not easily available in the public archive, which hampered our ability to conduct fine-grained analysis of users. A further analysis of this archive would benefit from using the associated email data to more precisely identify users. Techniques described by Bird et al [3] could be used in support of this approach.

This paper presents a largely quantitative analysis of the SIGCSE Mailing Archive, which illuminates certain aspects of the community. However, a qualitative analysis could shed insight on other aspects. Deeper analysis of posters' word usage, habits, and language could reveal potentially interesting information about the SIGCSE community. Proper qualitative and phenomenological analysis, such as that conducted by Anders et al [2], would be even more illuminating.

10 CONCLUSIONS

We have characterized the SIGCSE Members mailing list in a number of ways. We have performed some basic topical analysis to describe trends in discussion themes. Posts and posters were analyzed to derive email and thread characteristics, including from a temporal perspective. We report a successful application of machine learning to classify posts as job ads and conference discussion, and an unsuccessful attempt to predict thread popularity using statistical models. We hope that this reveals some deeper insight about the SIGCSE community, and gives the society an opportunity to reflect on where it has come in the past 20 years.

REFERENCES

[1] Izzat Alsmadi and Ikdam Alhami. 2015. Clustering and classification of email contents. *Journal of King Saud University-Computer and Information Sciences* 27, 1 (2015), 46–57.
[2] Anders Berglund and Raymond Lister. Debating the OO Debate: Where is the Problem?. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88 (Koli Calling '07)*. 171–174.
[3] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, and Anand Swaminathan. Mining email social networks. In *Proceedings of the 2006 international workshop on Mining software repositories*. 137–143.
[4] Kim B Bruce. 2004. Controversy on how to teach CS 1: a discussion on the SIGCSE-members mailing list. In *ACM SIGCSE Bulletin*, Vol. 36. 29–34.
[5] Curtis R. Cook. CS0: Computer Science Orientation Course *(SIGCSE '97)*. 87–91.
[6] Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. *Machine learning: ECML 2004* (2004), 217–226.
[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[8] Samuel Rebelsky and William J. Turner. 2017. Mailing Lists. http://sigcse.org/sigcse/membership/mailing-lists. (2017). Accessed: 2017-08-18.
[9] Seongwook Youn and Dennis McLeod. 2007. A comparative study for email classification. *Advances and innovations in systems, computing sciences and software engineering* (2007), 387–391.