Evaluating a Computational Approach to Labeling Politeness: Challenges for the Application of Machine Classification to Social Computing Data

ERIN R. HOFFMAN, Human Centered Design & Engineering, University of Washington
DAVID W. MCDONALD, Human Centered Design & Engineering, University of Washington
MARK ZACHRY, Human Centered Design & Engineering, University of Washington

Social computing researchers are beginning to apply machine learning tools to classify and analyze social media data. Our interest in understanding politeness in an online community focused our attention on tools that would help automate politeness analysis. This paper highlights one popular classification tool designed to score the politeness of text. Our application of this tool to Wikipedia data yielded some unexpected results. Those unexpected results led us to question how the tool worked and its effectiveness relative to human judgment and classification. We designed a user study to revalidate the tool with crowdworkers labeling samples of content from Wikipedia talk pages, imitating the original efforts to validate the tool. This revalidation points to challenges for automatic labeling. Based on our results, this paper reconsiders politeness in online communities as well as broader trends in the use of machine classifiers in social computing research.

CCS Concepts: • **Human-centered computing → Collaborative and social computing → Collaborative and social computing systems and tools • Computing methodologies → Machine learning**

KEYWORDS

Social Computing; Politeness; Machine Classification

ACM Reference format:

Erin R. Hoffman, David W. McDonald, and Mark Zachry. 2017. Evaluating a Computational Approach to Labeling Politeness: Challenges for the Application of Machine Classification to Social Computing Data. *Proc. ACM Hum.-Comput. Interact.*, 1 (2). 10.1145/3134687
https://doi.org/10.1145/3134687

1 INTRODUCTION

Ashley was working on writing some code that relied on a software library that she had not used before. Her code was not working as expected and she began to suspect that the library was generating some unexpected result. She was certain that someone must have had a similar problem before, so she posted a question on a bulletin board support system for this library. The answers she got back ranged from "RTFM" to other responses that seemed somewhat abusive. Ashley wondered whether this particular help channel was going to be all that helpful.

Author's email: Erin R. Hoffman <erinrhof@uw.edu>, David W. McDonald <dwmc@uw.edu>, Mark Zachry <zachry@uw.edu>.

DOI: 10.1145/3134687

Posts in online communities and other textual communications are frequently misinterpreted. The verbal and behavioral cues that help us understand and interpret a given piece of communication are not present in most textual messages. Norms of behavior can vary across online communities and understanding how different communities behave is one important challenge that faces participants as well as community managers.

For example, whether a given message is polite or impolite is difficult to discern in the most general sense because short messages may not have enough information to gauge the politeness, whereas long messages can include multiple and even competing motivations, yielding mixtures of politeness. This has created barriers in understanding how politeness impacts social media as well as computer supported cooperative work (CSCW) subject areas such as online collaboration and peer production.

One way to scope this problem is to consider situations where one person is making a request of another. In the scenario above, Ashley posts a question, asking for some kind of help or information about the software library. Requests for help, requests for information, and requests for specific actions, are but a few of the types of requests that we make of each other, both in face-to-face communication and in online communities. Such requests, it is broadly assumed, are likely to be embedded in notably polite terms, helping ingratiate the requestor with the person in a position to honor the request.

In this paper we reconsider a prominent machine learning classification tool that is designed to score a piece of text on a numeric scale that indicates whether the text is impolite, neutral, or polite. We did not build this tool, but attempted to apply it to understand one online community. However, in our effort to apply the tool we identified some anecdotally indeterministic behavior. This questionable behavior provoked our exploration to understand the characteristics of the tool more fully.

The specific contributions of this paper are three-fold. First, we describe a revalidation of the tool through a carefully constructed user study. This revalidation was designed to be similar to the original efforts to validate the tool, but differs in some specific ways. Second, given the results of the revalidation, we reconsider the prior work related to politeness in an attempt to reframe a trajectory for studying politeness in an online community. Third, by making visible some performance challenges of this example Machine Learning (ML) based classification tool, we reflect on the growing interest in the social computing community for leveraging ML tools to classify and analyze data.

In the following sections we outline related work that has used machine classification techniques to identify and label politeness in online communities. We highlight a specific classification tool that has grown in popularity and some of the challenges we had with it. We then describe our methods and data analysis for a user study that we designed to revalidate the tool and report the results of this revalidation. In the closing sections of the paper we reconsider some of the foundational work in conversational politeness and discuss our results in the context of a trend in social computing research to apply machine classifiers to understand and analyze social computing data.

2 POLITENESS IN SOCIAL COMPUTING RESEARCH

Many social computing researchers have approached the issue of measuring politeness in online communities, both to understand the communities as they currently exist and to potentially develop interventions. In the context of understanding online communities, researchers have explored the potential impacts of politeness when studying Pinterest [15], Instagram [2], Facebook [33], gendered differences [7], and newcomers to online communities [30, 31]. In the context of developing interventions, researchers have explored using positive politeness to promote useful discourse in online comments [10], evaluate contributions on the open source repository platform GitHub [32], promote teamwork behavior [22], and improve team decision-making [14]. With respect to politeness, all of these studies relied on qualitative methods, such as utilizing human scorers to label text samples or interviewing members of online communities.

Within CSCW research, these qualitative methods have proven valuable for understanding online communities, but researchers have repeatedly expressed a desire for tools to automate politeness labeling. For example, Burke and Kraut [6] utilized hundreds of human scorers to study the role of politeness in

eliciting responses on online forums. They identified differences in the most successful requests based on forum topic (e.g., less polite posts were more successful in economics groups, while more polite posts were more successful in C programming groups). Based on this outcome, they suggested creating an ML based tool "that can be applied to much larger corpora, for greater generalizability" as well as for the "design of automatic interventions, such as a 'politeness checker' that suggests linguistic strategies to newcomers before they post their first messages." Similarly, after interviewing journalists and participants in online news forums, Diakopoulos and Naaman [10] suggested creating "advanced filtering tools ... based on the politeness of comments" in order to help manage "individual differences in reading motivations." Additionally, an automated tool for identifying impolite language could potentially help reduce the cost of expensive commercial content moderation for online communities seeking to provide a more welcoming environment for newcomers [18]. Thus there exists a growing demand for an automated politeness labeler, both for understanding online communities and for developing interventions.

In 2013 Danescu-Niculescu-Mizil et al. [9] from the Stanford Natural Language Processing Group published a paper detailing the creation of an ML based politeness tool to answer this call for help. They first created a corpus of over 10,000 requests from Wikipedia discussions. They annotated 4353 requests with politeness scores using Amazon Mechanical Turk (MTurk). Five different MTurk workers each indicated the politeness of sets of 13 requests with a slider input of discrete scores between -3 meaning "very impolite" through 0 meaning "neutral" through +3 meaning "very polite" (the workers only saw categorical labels, and not the numerical values). They then trained a linguistically informed SVM classifier to predict the politeness assigned by MTurk workers, using a logistic regression model to generate a predicted politeness score between 0 and 1. They validated the classifier by comparing its scores to those generated by another set of samples also scored by MTurk workers. Analyzing the output from the tool, they set cutoff scores for "impolite," "neutral" and "polite" labels with "impolite" requests having scores ranging from 0 to 0.25, "neutral" requests having scores ranging from 0.25 to 0.75, and "polite" requests having scores ranging from 0.75 to 1.

They demonstrated the utility of these categories by making claims about politeness in the online Q&A site StackExchange and how it differed from group to group (e.g., Python programmers vs. Ruby programmers) and between individuals (e.g., StackExchange users with low vs. high reputations). The paper authors concluded that "by providing automatic politeness judgments for large amounts of new data on a scale unfeasible for human annotation, [a system] can also enable a detailed analysis of the relation between politeness and social factors." Since the publication of this paper, the authors have also open-sourced the tool by publishing the source code along with example input on GitHub. At this time, dozens of papers have cited Danescu-Niculescu-Mizil et al., several of which utilized the tool as part of their analysis.

3 REVALIDATING A POLITENESS LABELING TOOL

Our research team became interested in the Danescu-Niculescu-Mizil et al. politeness scoring tool because it had been trained on Wikipedia data and has been used in numerous studies. We began by applying the tool to some text samples from Wikipedia talk pages and checking the labels by hand. At this point, our perceptions about the scoring relative to the text samples caused us some concern. We manually attempted to tweak samples and re-score them in order to make them more polite or more impolite as a manual test of the tool's validity, but the score differences between the original and modified samples often seemed unpredictable. These fluctuations made us especially uncertain about the "impolite," "neutral," and "polite" ranges described by the original paper. We are not the first to have concerns regarding the performance of the tool. Our concerns mirrored those expressed by Kumar [21]. We therefore decided to revalidate the politeness tool to determine whether our concerns were realistic.

In the following sections we outline this revalidation process. We briefly describe two different ways to apply the tool then describe how we specifically utilized it in the revalidation. We gathered a new dataset from Wikipedia and used the politeness tool to assign scores between 0.0 and 1.0 to these samples. We then had Amazon Mechanical Turk crowdworkers label samples as impolite, neutral, or polite. Using scores from the tool and labels from the MTurk workers, we performed a statistical analysis to understand the

relationship between the two. This forms a type of revalidation of the tool and can serve to make visible some aspects of the way the tool works.

3.1 Working with the ML Labeling Tool

The tool can be used in two different ways and has been used in both ways across the prior literature. For both forms of use, it is important that the text include a "request". This means the text should have some type of question - or at least include a question mark. Another important caveat is that the tool was trained

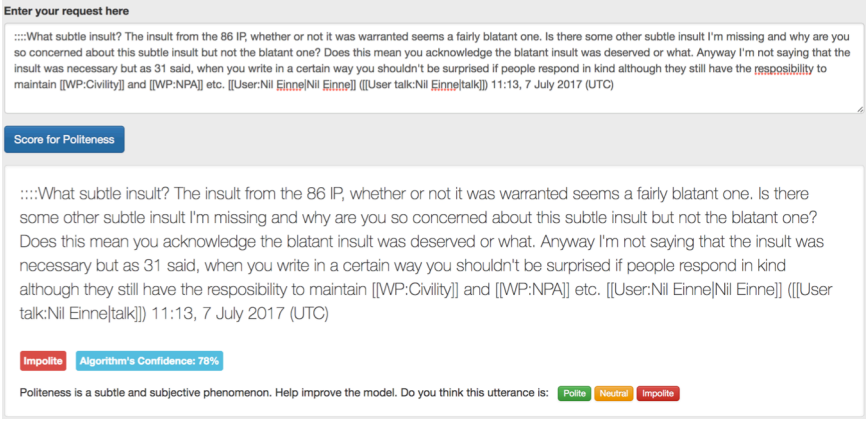

Figure 1. Example of the web based use of the tool on text from the Wikipedia main page talk page

on and appears to perform better when the text is sentence pairs. Longer blocks of text are more likely to have mixed intent and can challenge the performance of the tool. One way to apply the tool is through a web interface[1] that will label chunks of text pasted into the web page as “impolite,” “neutral,” or “polite”. A sample labeling of text from the talk page of the Wikipedia main page is provided in Figure 1 to illustrate the web interface to the tool.

But also, the source code for the tool is available[2]. The code was revised to be Python 3 compliant in March 2017, but that version was not available when we began this work, and further the Python 3 version had not been regression tested against the prior working version as of July 2017. We instead used the v1.01 (October 2014) version of the source code for our work with the tool. One advantage of using the source code version is that one could train a new model using new data. Since most social computing researchers are likely to use the tool "off-the-shelf" we employed the tool that way. Once samples have been collected, and cleaned of extraneous markup, sentence pairs are passed through a tokenizer and through the Stanford CoreNLP parser to generate text dependencies, which are stored in a JSON file. This JSON is what is passed into the politeness tool. The tool then provides a 0.0 to 1.0 floating point value score for the sentence pair. Once a sentence pair is scored, the value of that score can be used to label the pair as "polite", "neutral" or "impolite."

3.2 Gathering Data from Wikipedia

Using the Wikipedia API, our team collected 91,600 revisions from Wikipedia talk pages; article talk, user talk, and WikiProject talk. Distinct from the encyclopedia-style pages that Wikipedia is known for, talk pages are the spaces on Wikipedia where users can discuss page content and coordinate their editing efforts. Every article page on Wikipedia has an associated talk page. Additionally, all user pages (essentially user profiles) and WikiProject pages (pages centered around collaborative groups of Wikipedia editors) have

1. http://politeness.cornell.edu/
2. https://github.com/sudhof/politeness/

associated talk pages. Because the Danescu-Niculescu-Mizil et al. paper states that they used talk pages but does not specify what types of talk pages were included in their original training data, we sampled from all three spaces, starting from the talk pages with the most content according to the Wikipedia Database Reports. We used the Python library difflib to parse these revisions into individual edits.

3.3 Calculating Politeness Scores

Danescu-Niculescu-Mizil et al. specify that their tool was designed for sentence pairs, rather than text samples of arbitrary length, so we pared down the 91,600 revisions to 54,047 sentence pairs. Not every user revision contains an appropriate and legitimate sentence pair. After cleaning wikitext markup and tokenizing, we ran the samples through the Stanford CoreNLP parser to create the phrase structure tree JSON required by the politeness tool, and then scored these 54,047 samples using the politeness tool.

The paper also indicates that the politeness scoring is designed to work on "requests". Included in the sample code is a script that implements a “heuristic to determine whether a document looks like a request.” For each of our 54,047 sentence pairs we applied the heuristic code so that we had both a real value score and a binary marker as to whether the sentence pair contained a request or not.

3.4 Amazon Mechanical Turk Scoring

The original training data for the politeness classifier was created using Mechanical Turk (MTurk). In the original, MTurk workers were asked to score sentence pairs containing requests using a sliding scale input from -3 "very impolite" to +3 "very polite". In the original every sentence pair was scored 5 times. An "attentiveness task" facilitated a check of the individual MTurk workers’ scores. In the original, a worker z-score was used to normalize scores.

We recognize how difficult it is to judge politeness. For our revalidation task, instead of using the exact same seven point scale (-3 through 0 to 3; very impolite to very polite) we decided to try to simplify the labeling task using just "polite", "neutral", and "impolite". Our task directly mirrors the suggested labels derived from the tool based on the tool’s floating point scores. Our task interface did not attach these labels to any numerical values. Additionally, since the tool was designed to work specifically with requests, we asked the MTurk workers to judge if each sentence pair contained a request. In our later analysis, we consider separately their overall responses and responses when workers considered the sample to contain a request.

We generated a set of target sentence pairs by sampling from our 54,047 sentence pairs. Since the tool provides floating point scores, we leveraged those scores to generate a more balanced set of target sentences. Random sampling would have generated a highly unbalanced sample of targets that would likely be mostly “neutral.” We used a stratified sampling to generate target sentences across all of the categories, oversampling at the tails to make sure there were "polite" and "impolite" sentence pairs in our set. Our sample consisted of 90 sentence pairs, which we then had labeled by MTurk crowdworkers. In order to reduce the potential for ordering effects, we randomly selected 8 sentence pairs for each crowdworker task.

As a check on MTurk worker quality, we used a set of gold standard sentence pairs whose labels were based on unanimous preliminary labeling. Those labels were determined by a small set of colleagues who were not associated with this project. Out of 32 total MTurk workers, 26 correctly labeled the gold standard sentences. After eliminating answers from workers who were inaccurate on the gold standard sentences, we were left with 1,387 MTurk worker labels of the 90 sentence pair samples. Each individual sentence pair was scored by between 4 and 26 unique MTurk workers.

Table 1. Descriptive statistics of the revalidation

	Samples Labeled Impolite	Samples Labeled Neutral	Samples Labeled Polite
Politeness Tool (Original Paper Thresholds)	42.0%	13.0%	45.0%
MTurk Labeling (All Samples)	12.9%	53.4%	33.7%
MTurk Labeling (Requests Only)	7.8%	39.1%	53.1%

4 REVALIDATION RESULTS

4.1 Descriptive Statistics of MTurk Worker Categorizations

Of all of the samples categorized by MTurk workers, 48% were labeled requests. Because the tool was designed to work for requests but has also been applied to non-requests in many subsequent studies, we will analyze both the full dataset and the subset that were labeled as containing requests.

For the full dataset, workers labeled the majority (53.4%) of samples as "neutral," while for the request-only dataset, workers labeled the majority (53.1%) of samples as "polite." Based on the thresholds of 0.25 and 0.75 defined in the original paper, the plurality of samples were expected to fall within the "polite" range. Table 1 describes these descriptive statistics in more detail, and Figure 2 provides this same information in a visual format to help compare the original thresholds, the MTurk labels of all samples, and the MTurk labels of request samples only.

4.2 Inter-Rater Reliability

We calculate inter-rater reliability with some caveats that relate to the way our labels were collected. We considered "impolite," "neutral" and "polite" to form an ordinal scale. Using MTurk we had multiple raters and missing values, since not all MTurk workers rated all sentence pair samples. Given that, the most appropriate measure of inter-rater reliability was Krippendorff's alpha [20]. The resulting alpha was 0.445, as calculated by Deen Freelon's ReCal OIR tool [13]. This approximately means that 44.5% of the sentence pairs were coded with a degree of agreement better than random chance.

Interpreting this alpha requires some nuance. Further, comparing our inter-rater reliability relative to

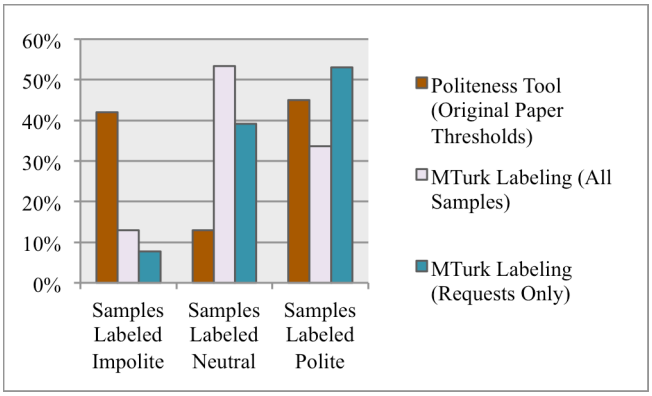

Figure 2. Visual snapshot of descriptive statistics

that of the original labeling task in Danescu-Niculescu-Mizil et al. is not a straightforward analysis. Krippendorff himself raises the question "What is an acceptable level of reliability?" and answers with "Unfortunately, although every content analyst faces this question, there is no set answer." Therefore we will summarize the inter-rater reliability analysis performed in the original paper, followed by our conceptually similar analysis.

Danescu-Niculescu-Mizil et al. did not report an alpha or kappa representing their inter-rater reliability, and explicitly defended their decision not to use a Cohen's or Fleiss's kappa because their work violated the assumptions of those statistics. Instead, they calculated the mean inter-annotator pairwise correlation (which was not reported) and demonstrated that it was non-random by comparing it to the correlations of randomly-generated sets of scores. Specifically, they performed a Wilcoxon signed rank test comparing the mean pairwise correlation of their MTurk worker scores to a randomized set of scores, and found a $p<0.0001$. This means that their MTurk workers' scores correlated such that there is less than 0.01% likelihood that the correlation was random.

We therefore performed a conceptually similar test to demonstrate that our inter-rater reliability was non-random. We compared our observed Krippendorff's alpha to the hypothetical Krippendorff's alphas found in a large number of randomly generated sets of labels. Specifically, we performed a stratified Monte Carlo permutation test [12]. A true permutation test would compare our alpha to the alphas generated by every possible permutation of each MTurk worker's labels, but this proved computationally expensive because it would require the calculation of 6.55 x 1029 alpha values. Therefore, instead we performed a Monte Carlo permutation test, randomly shuffling the labels 100,000 times and comparing the resulting alphas to our observed alpha.[3] Zero of the hypothetical random alphas were greater than our alpha, resulting in $p < 0.00001$. This means that there is less than 0.001% likelihood that our alpha was random. Thus our MTurk workers performed similarly to or better than those in the original study.

4.3 Results of Multinomial Logistic Regression

In addition to the descriptive analysis, which appears to show a difference between the politeness tool and the MTurk results (with or without the request-only requirement), we performed an inferential analysis to compare the scores assigned by the politeness classifier to the category labels assigned by the MTurk workers. Because this study compared nominal outcomes (categories of "impolite," "neutral" and "polite") to a predictor variable (the score output from the politeness classifier tool), we used a multinomial logistic regression. This generated a probability curve for each label, where the x-axis is the score output (between 0 and 1) from the politeness tool, and the y-axis is the probability that the sample will be assigned that label. Figure 3 shows the ideal or expected curves generated by the thresholds of 0.25 and 0.75 defined in the original paper. It would be very difficult to get this idealized performance on such a subjective labeling task, so we present these for comparison when considering the actual probability performance. Figure 4 shows the observed curves (based on MTurk labels) for all samples, while Figure 5 shows the observed curves for only samples labeled as containing requests.

4.3.1 Thresholds between Impolite, Neutral, and Polite. As previously stated, the Danescu-Niculescu-Mizil et al. paper described "impolite" samples as having scores between 0 and 0.25, "neutral" samples as having scores between 0.25 and 0.75, and "polite" samples as having scores between 0.75 and 1. If the categories as understood by the MTurk workers in our study matched these categories (as represented in Figure 3), we would expect:

- The "impolite" curve to have the highest y value (probability) at x=0 (the lowest possible score from the politeness tool)

3. While we used Deen Freelon's ReCal OIR tool to calculate the observed Krippendorff's alpha noted previously, this strategy was not scalable for the permutation test, since each calculation with ReCal OIR requires uploading a CSV file to a web interface. Rather than uploading thousands of CSVs, we instead used a Python implementation of Krippendorff's alpha by Lee Becker that is hosted on GitHub [3]. Both Freelon's ReCal OIR and Becker's Python implementation produced the same alpha for our observed MTurk data, which we believe validates their performance.

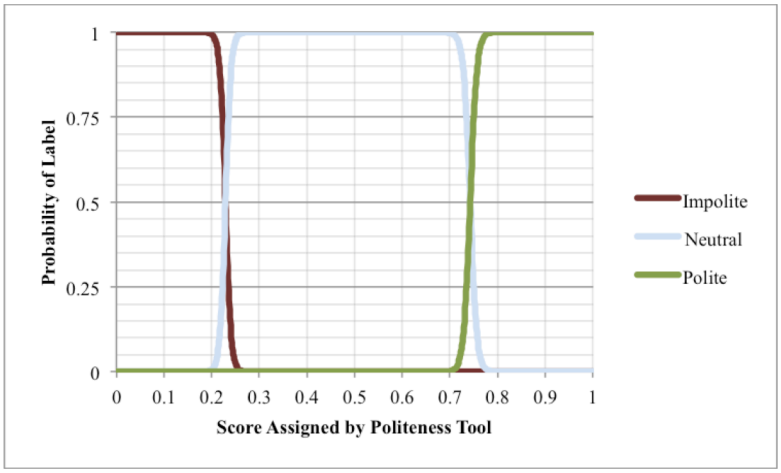

Figure 3. Idealized probability curves for labels given thresholds of 0.25 and 0.75 specified in paper

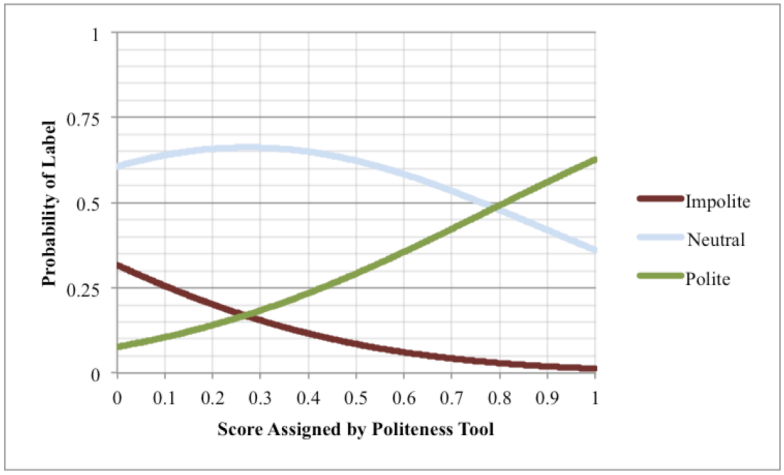

Figure 4. Label probability curves for all MTurk samples

- The "neutral" curve to surpass the "impolite" curve and have the highest y value after x=0.25 and drop off significantly after x=0.75
- The "polite" curve to surpass the "neutral" curve and have the highest value after x=0.75
- The "polite" curve to have the highest y value at x=1 (the highest possible score from the politeness tool)

As Figures 4 and 5 demonstrate, our results do not match the thresholds defined in the original study. In both cases, the "neutral" curve has the highest y value (probability) at x=0, not the "impolite" curve. This means that a threshold between "impolite" and "neutral" simply doesn't exist, because the MTurk workers were more likely to assign the category of "neutral" than the category of "impolite" no matter what score the politeness tool assigned. With the full dataset (Figure 4), the threshold between "neutral" and "polite" is

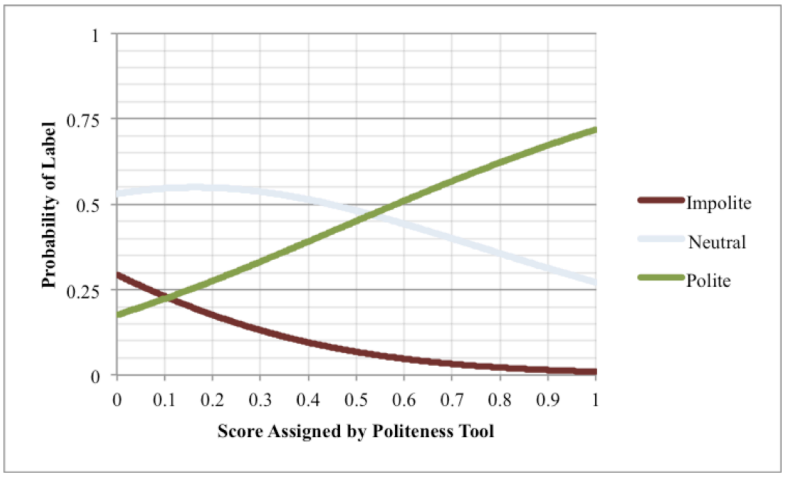

Figure 5. Label probability curves for MTurk samples labeled "request" only

relatively close to the value described in the original study paper. It sits at x=0.789 rather than x=0.75.[4] However in the request-only dataset, which theoretically should align more closely with the original thresholds, "polite" surpasses "neutral" in probability at x=0.531, a number much further from their threshold. A summary of the new thresholds is shown in Table 2.

4.3.2 Category Probabilities Across the Score Spectrum. While the threshold concept describes the category probabilities in terms of the x-axis (the politeness score at which one label becomes more likely than another), it leaves out the y-axis (the probability values themselves).

Considering the probability more carefully, when "polite" surpasses "neutral" (in the full dataset) the probability of the original thresholds agreeing with human judgment is only 48.5%. That probability is not even equivalent to flipping a coin. That is, the agreement of human judgment with the original thresholds when the tool scores the sentence at about x=0.789 is no better than a random coin toss. This distinction is important in understanding that just because "polite" is the most likely label above x=0.789, that threshold is still very close to a random guess between "polite" and "neutral" rather than a definitive assertion that a

Table 2. Ranges where Impolite, Neutral, and Polite labels are more likely given method of labeling

	Impolite Label Most Likely	Neutral Label Most Likely	Polite Label Most Likely
Politeness Tool (Original Paper Thresholds)	0 to 0.25	0.25 to 0.75	0.75 to 1
MTurk Labeling (All Samples)	N/A	0 to 0.789	0.789 to 1
MTurk Labeling (Requests Only)	N/A	0 to 0.531	0.531 to 1

4. This use of the multinomial logistic regression violates the assumption of independent samples (because we had MTurk workers score multiple samples rather than having a separate worker for each sample), so we used bootstrapping (1000 replications) to generate confidence intervals. For x=0.789, the point at which the "polite" probability curve surpasses the "neutral" probability curve in the full dataset, the 95% CI was from 0.737 to 0.841. For x=0.531, the point at which the "polite" probability curve surpasses the "neutral" probability curve in the request only dataset, the 95% CI was from 0.436 to 0.626. For x=0.268, the point at which the "polite" probability curve surpasses the "impolite" probability curve in the full dataset, the 95% CI was from 0.221 to 0.315. For x=0.107, the point at which the "polite" probability curve surpasses the "impolite" probability curve in the request only dataset, the 95% CI was from 0.089 to 0.196.

Table 3. Probabilities of labels at thresholds for the full dataset; 0.268 is where the Polite curve surpasses the Impolite curve and 0.789 is where the Polite curve passes the Neutral curve

	Score = 0	Score = 0.268	Score = 0.789	Score = 1
Probability Labeled Impolite	31.7%	16.9%	3.0%	1.3%
Probability Labeled Neutral	60.6%	66.2%	48.5%	36.0%
Probability Labeled Polite	7.7%	16.9%	48.5%	62.7%

sample scored in this range will necessarily be interpreted as "polite." When "neutral" is the most likely label, it does not mean that zero MTurk workers would label those samples as "impolite." In the full dataset, at x=0 (most impolite possible score) there is a 60.6% chance of an MTurk worker labeling a sample "neutral" and only a 31.7% chance that they would label it "impolite." In the request-only dataset, at x=0 there is a 53% chance of a sample being labeled "neutral" and a 29.4% chance of it being labeled "impolite." A broader set of these probability values can be seen in Table 3 (all samples) and Table 4 (samples labeled as containing requests only).

5 ANALYSIS

Overall, the tool does not behave in the way most users of the tool would expect. The results demonstrate that this tool is somewhat effective at scoring "polite" text, but is ineffective at scoring "impolite" and "neutral" samples. As well, the probability that the category scores correspond to human judgment is in serious question.

5.1 Applying This Tool in Studying Online Communities

The tool is somewhat effective at distinguishing "polite" samples from other samples given a relatively high score from the tool. This aligns well with humans who are likely to assign the "polite" label over any other label when the tool's score is above 0.789. Considering only samples labeled as requests, humans are more likely to assign the "polite" label over any other label when the tool's score is above 0.531. Thus, making sure that the tool is scoring a request seems to improve the correspondence of the tool score to human labeling. At the tool's maximum score of 1, humans are either 62.7% or 71.9% likely to assign the "polite" label for all samples and requests respectively. The increasing probabilities at the "polite" end are unsurprising given that the linguistic theory underpinning this tool focuses on politeness strategies (how people intentionally communicate politeness) [5]. These strategies are not effective as one becomes impolite and the tool apparently suffers from this specific bias.

Table 4. Probabilities of labels at thresholds for request data only; 0.107 is where the Polite curve surpasses the Impolite curve and 0.531 is where the Polite curve passes the Neutral curve

	Score = 0	Score = 0.107	Score = 0.531	Score = 1
Probability Labeled Impolite	29.4%	22.6%	6.0%	1.0%
Probability Labeled Neutral	53.0%	54.8%	47.0%	27.1%
Probability Labeled Polite	17.6%	22.6%	47.0%	71.9%

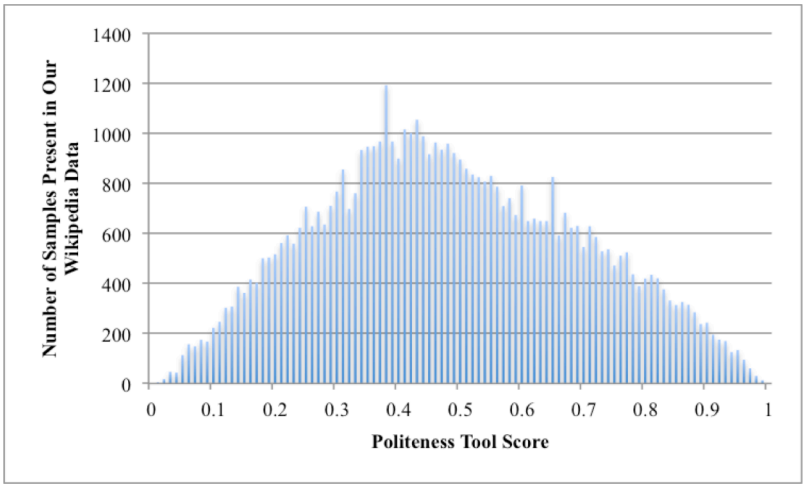

Figure 6. Histogram of politeness scores for all 54,047 sentence pairs in our Wikipedia dataset, illustrating the very small fraction of the total data that could be reliably labeled

Another consideration is how many samples (sentence pairs) from the entire set could be somewhat accurately scored. After conducting this revalidation, we can ask the question: Given scores generated by the tool, what percentage of data could possibly be scored reliably? We scored all of our 54,000 sentence pairs from Wikipedia and note that scores in the highest politeness ranges represent a surprisingly small fraction of the sentences written and collected. For example, let's say we want at least a 50% probability (coin toss) that a sentence pair would be labeled "polite" by a human. That probability corresponds to a threshold score of 0.811 or higher by the tool (95% CI 0.762 to 0.860). At this score/probability only 7% of the 54,000 samples would be considered accurately scored. If we raise the bar and demand at least a 60% probability that a sample would be labeled as "polite" by a human, that corresponds to a threshold score of 0.959 by the tool (95% CI 0.908 to 1.01), which covers only 0.3% of the data. See Figure 6 for the distribution of scores in our Wikipedia sample. This severely challenges the assumptions made by most of the prior work that the tool can effectively be used to characterize the overall politeness of a given online community.

However, given an extremely large dataset, 60% accuracy on 0.3% of a sample may still be enough to allow researchers to productively use this tool to study online communities. They just should not characterize and profile the entire community with this tool. *If the research were framed around explicit use of politeness strategies, this tool could be used to identify candidate sentences that should probably be analyzed by hand to understand how politeness was being used. A sufficiently large corpus could still yield an interesting politeness analysis through mixed methods.*

5.2 Applying This Tool in Interventions

This tool is ineffective at distinguishing between "impolite" samples and non-"impolite" ("neutral" or "polite") samples. Even at a score of 0, the lowest possible output from the politeness tool, humans are more likely to label samples as "neutral" than "impolite." *This means that applying this tool in contexts where impoliteness is the main focus, such as automated content moderation to make online spaces more welcoming, would be inappropriate.*

More broadly, this tool should not be used to make claims about individual text samples. A very high or a very low score by the tool has includes significant uncertainty relative to how that sample would be labeled by a human. This tool would not be an appropriate way to reward users for extra-polite comments or to make claims about how tweaking the politeness of a request online might impact the responses received by that request.

6 DISCUSSION

Machine learning and computational linguistic approaches are very appealing to those of us who study online communities. They promise to provide rich data at an unprecedented scale. Sometimes, however, the tools do not live up to that promise. *Part of the challenge, we surmise, may be attributed to the translation of such approaches from other fields to our own.* For example, research contributions in computational linguistics are often based on clean, uniform data. The messy, in the wild data from online communities is necessarily difficult to process and analyze.

An additional challenge for social computing research is inherent in borrowing theories of social behavior (e.g., expressions of politeness) from other fields. *When theories that have been developed in a field are decontextualized from the norms of the community in which they originate, much is lost.* Politeness theory, for example, has its origins in linguistic anthropology. Though widely known and influential, the theory itself is subject to challenges and critiques that have conditioned its use in the linguistics community. The thrust of these critiques and how they have led to more nuanced ways of using theory or conditioning results when the theory is employed by linguists today is nearly always lost when politeness theory is borrowed by other fields. When social computing researchers, for example, attempt to use the theory to support their work, such as developing a labeling technique in support of machine learning goals, the borrowing inevitably relies on a basic model and set of defined rules. Such borrowing likely misses many of the as-yet-unaddressed challenges and recognized-but-only-partially-answered gaps in politeness theory. Within linguistics, for example, there are well-known challenges to the theory, including cultural differences, variations caused by non-routine communication strategies, and insider-outsider language conventions. Such challenges are relevant to applications of politeness theories to online communities like Wikipedia where language norms develop in ways unique to the community (see, for example, [4]). For example, experienced Wikipedians are familiar with shorthand references to system policies and would not likely perceive a request to adhere to policy as impolite, whereas a newbie or outsider might take exception to being pointed to a policy referred to in an abbreviated form with little elaboration about the requestor's rationale. *Insider language perspectives like this matter a great deal for those in the system as they interact, but they are not accounted for at all in simple applications of politeness theory to the language scraped from an online community.*

Beyond this, computational social scientists might also be overestimating the power of tool-based approaches to identifying social intentions and behaviors because they underestimate the complexity of the task at hand; *just because something is the best computational approach to politeness published so far does not necessarily mean that this approach is ready to be used in the ways that people studying online communities desire.* Our approach in this study works toward exposing more about the performance of this classifier tool than most who have used the tool have attempted. Opening up the tool to make its performance more visible calls attention to how limited a tool can be for addressing specific questions of interest. The performance of the tool is somewhat blunt and is not for nuance in computer-supported interaction. *In short, we contend that the things that can be accomplished via ML do not easily translate into the kind of questions that are asked in the social computing research community.*

The gap between theory and application in computational linguistics is large. Additional systematic and deep study might help us understand this gap and how we might bridge it. One of many difficult challenges for the creation of ML tools for subjective constructs like "politeness" is the issue of getting agreement from human raters on how these constructs should be defined. Our low Krippendorff's alpha is one example of this issue. Another concrete example is that of understanding harassment [17]. Guberman and Hemphill set out to create an ML-based anti-harassment content moderation tool, but ultimately failed to reach adequate levels of inter-rater agreement on what constituted harassment, due to "rarity, context, and individual coder's differences." *While we are confident that it would be possible to create a politeness classifier that performed in a better, more nuanced way, doing so would require a much more complex rule set and model of language-use context.*

Such challenges, however, are not widely acknowledged by researchers in our field. As we considered how this politeness classification tool was being used by others, we noted that once the Danescu-Niculescu-Mizil

et al. paper was published and the code was released, researchers immediately began utilizing it to study and make research claims about online communities. Still others used the classifier in an integrated fashion to develop more complex tools, such as tools for content moderation. The numerous studies employing this tool include investigations of:

- how politeness impacts the success of asking for a favor [1],
- whether polite users are more productive on the software development discussion platform JIRA [13, 24, 25, 26, 28],
- how politeness predicts betrayal in online strategy games [23],
- the relationship between politeness and formality in online communities [29].
- the politeness of Polymath online research communities [19],
- tools for the coexistence of diverse opinions online [16],
- how politeness correlates with gender diversity on GitHub teams [27],
- how politeness impacts the success of questions asked on Reddit [8]

Additionally, at least 32 publications have used the corpus from the Danescu-Niculescu-Mizil et al. paper or referenced it as a positive example to support some other form of subsequent work.

Given our revalidation, the results of prior studies that have relied on the Danescu-Niculescu-Mizil et al. tool should be carefully reconsidered. In particular, studies that make claims about the neutral and impolite portions of the scale probably suffer from serious inaccuracy issues relative to what humans would actually say about those texts.

Many studies have employed the tool's web interface (see Figure 1). Unlike the open-sourced tool posted on GitHub, this interface does not require writing Python code or running a local instance of the Stanford CoreNLP parser, but rather allows users to submit for scoring any type of text string they desire. This interface does not have any mechanism for indicating whether the text submitted was indeed a request, although the tool was designed to work on requests. As we note in our revalidation, the tool agrees more with human labelers on the "polite" labels when the sentence pairs are identified as requests by human labelers. Also, the tool was designed to work on sentence pairs, and some of our anecdotal evidence suggests that it performs better on sentence pairs. Because there is no requirement that the text submitted via the web interface contain a request, and because there is no restriction on the amount of text pasted into the web interface, the results of several of these previous studies are open to additional noise in the reported results.

7 CONCLUSION

This study focused on a particular tool that attempts to automatically categorize statements as impolite, neutral, or polite. Our struggles with the tool caused us to reconsider the reliability of the tool relative to human raters, leading us to a revalidation of the tool that strongly mirrors how the original tool was validated. Our revalidation identified some cutoffs that were similar to the cutoffs identified in the original validation. Our results call into question the utility of the tool for broadly addressing politeness in social computing data. Our discussion identified a set of challenges for automated labeling of social computing data. With additional research, these methods might have broad applicability for researchers and designers; however, individuals wishing to apply these tools in their current state of development should fully understand their limits.

8 ACKNOWLEDGEMENTS

The authors thank the reviewers for their thoughtful comments during the review process. We also acknowledge numerous conversations with colleagues regarding the difficult challenges of detecting harassment, bullying and politeness in online communities. This work was supported in part by National Science Foundation (NSF) grant, IIS-1162114.

9 REFERENCES

[1] Tim Althoff, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2014. How to Ask for a Favor: A Case Study on the Success of Altruistic Requests. In Eighth International AAAI Conference on Weblogs and Social Media.

[2] Saeideh Bakhshi, David A. Shamma, and Eric Gilbert. 2014. Faces engage us: Photos with faces attract more likes and comments on Instagram. In Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14), 965-974.

[3] Lee Becker. 2013. Inter Annotator Agreement. GitHub repository. Retrieved July 2017 from: https://github.com/leebecker/InterAnnotatorAgreement

[4] Emily M. Bender, Jonathan T. Morgan, Meghan Oxley, Mark Zachry, Brian Hutchinson, Alex Marin, Bin Zhang, and Mari Ostendorf. 2011. Annotating social acts: Authority claims and alignment moves in Wikipedia talk pages. In Proceedings of the Workshop on Languages in Social Media, Association for Computational Linguistics, 48-57.

[5] Penelope Brown and Stephen C. Levinson. 1987. Politeness: Some universals in language usage. Cambridge University Press.

[6] Moira Burke and Robert Kraut. 2008. Mind your Ps and Qs: the impact of politeness and rudeness in online communities. In Proceedings of the 2008 ACM Conference on Computer Supporter Cooperative Work (CSCW '08). 281-284.

[7] Gina Masullo Chen and Zainul Abedin. 2014. Exploring differences in how men and women respond to threats to positive face on social media. Computers in Human Behavior, 38, 118-126.

[8] Yogesh Dahiya and Partha Talukdar. 2016. Discovering Response-Eliciting Factors in Social Question Answering: A Reddit Inspired Study. Director 24196(3295), 13-61.

[9] Christian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A Computational Approach to Politeness with Application to Social Factors. In 51st Annual Meeting of the Association for Computational Linguistics, 250-259.

[10] Nicholas Diakopuolos and Mor Naaman. 2011. Towards quality discourse in online news comments. In Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work (CSCW '11), 133-142.

[11] Giuseppe Destefanis, Marco Ortu, Steve Counsell, Michele Marchesi, and Roberto Tonelli. 2015. Software development: do good manners matter? (No. e1892). PeerJ PrePrints.

[12] Michael D. Ernst. 2004. Permutation methods: a basis for exact inference. Statistical Science, 19(4), 676-685.

[13] Deen Freelon. 2013. ReCal OIR: Ordinal, interval, and ratio intercoder reliability as a web service. International Journal of Internet Science, 8(1), 10-16.

[14] Erin Friess. 2011. Politeness, time constraints, and collaboration in decision-making meetings: A case study. Technical Communication Quarterly, 20(2), 114-138.

[15] Eric Gilbert, Saeideh Bakhshi, Shuo Change, and Loren Terveen. 2013. I need to try this?: a statistical overview of Pinterest. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2427-2436.

[16] Catherine Grevet. 2016. Being Nice on the Internet: Designing for the Coexistence of Diverse Opinions Oneline, PhD. diss., School of Interactive Computing, Georgia Institute for Technology, Atlanta, GA.

[17] Joshua Guberman and Libby Hemphill. 2017. Challenges in modifying existing scales for detecting harassment in individual tweets. In Proceedings of the 50th Hawaii International Conference on System Sciences, 2203-2212.

[18] Aaron Halfaker, Bryan Song, D. Alex Stuart, Aniket Kittur, and John Reidl. 2011. NICE: Social Translucence through UI intervention. In Proceedings of the 7th International Symposium on Wikis and Open Collaboration, 101-104.

[19] Isabel Mette Kloumann, Chenhao Tan, Jon Kleinberg, and Lillian Lee. 2016. Internet Collaboration on Extremely Difficult Problems: Research versus Olympiad Questions on the Polymath Site. In Proceedings of the 25th International Conference on the World Wide Web, 1283-1292.

[20] Klaus Krippendorff. 2004. Content analysis: An introduction to its methodology (2nd ed.). Sage Publications.

[21] Ritesh Kumar. 2014. Politeness in Hindi Online Texts: Pragmatic and Computational Aspects. PhD. diss., Centre for Linguistics, Jawaharlal Nehru University, New Delhi, India.

[22] Gilly Leshed. 2009. Automated language-based feedback for teamwork behaviors. PhD. diss., Cornell University, Ithaca, NY.

[23] Vlad Niculae, Srijan Kumar, Jordan Boyd-Graber, and Christian Danescu-Niculescu-Mizil. 2015. Linguistic Harbingers of Betrayal: A Case Study on an Online Strategy Game. In 53rd Annual Meeting of the Association for Computational Linguistics, 1650-1659.

[23] Marco Ortu. 2015. Mining software repositories: measuring effectiveness and affectiveness in software systems. PhD diss., Electronic and Computer Engineering, University of Cagliari, Cagliari, Italy.

[24] Marco Ortu, Bram Adams, Guiseppe Destefanis, Parastou Tourani, Michele Marchesi, and Roberto Tonelli. 2015. Are bullies more productive?: empirical study of affectiveness vs. issue fixing time. In Proceedings of the 12th Working Conference on Mining Software Repositories, 303-313.

[25] Marco Ortu, Guiseppe Destefanis, Mohamed Kassab, Steve Counsell, Michele Marchesi, and Roberto Tonelli. 2015. Would you mind fixing this issue?. In International Conference on Agile Software Development, 129-140.

[26] Marco Ortu, Guiseppe Destefanis, Steve Counsell, Stephen Swift, Michele Marchesi, and Roberto Tonelli. 2016. How diverse is your team? Investigating gender and nationality diversity in GitHub teams (No. e2285v1). PeerJ Preprints.

[27] Marco Ortu, Alessandro Murgia, Giuseppe Destefanis, Parastou Tourani, Roberto Tonelli, Michele Marchesi, and Bram Adams. 2016. The emotional side of software developers in JIRA. In Proceedings of the 13th International Workshop on Mining Software Repositories, 480-483.

[28] Ellie Pavlick and Joel Tetreault. 2016. An Empirical Analysis of Formality in Online Communication. Transactions of the Association for Computational Linguistics, 4(61-74).

[29] Vandana Singh, Aditya Johri, and Raktim Mitra. 2011. Types of newcomers in an online developer community. In Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work, 717-720.

[30] Hon Jie Teo and Aditya Johri. 2014. Fast, functional, and fitting: expert response dynamics and response quality in an online newcomer help forum. In Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, 332-341.

[31] Jason Tsay, Laura Dabbish, and James Herbsleb. 2014. Let's talk about it: evaluating contributions through discussion in GitHub. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, 144-154.

[32] Yi-Chia Wang, Moira Burke, and Robert Kraut. 2013. Gender, topic, and audience response: an analysis of user-generated content on Facebook. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 31-34.