Community-Driven Crowdsourcing: Data Collection with Local Developers

Christina Funk, Michael Tseng, Ravindran Rajakumar, Linne Ha
Google
1600 Amphitheatre Parkway, Mountain View, CA 94043
{christinafunk, michaeltseng, ravirajakumar, linne}@google.com

Abstract

We tested the viability of partnering with local developers to create custom annotation applications and to recruit and motivate crowd contributors from their communities to perform an annotation task consisting of the assignment of toxicity ratings to Wikipedia comments. We discuss the background of the project, the design of the community-driven approach, the developers' execution of their applications and crowdsourcing programs, and the quantity, quality, and cost of judgments, in comparison with previous approaches. The community-driven approach resulted in local developers successfully creating four unique tools and collecting labeled data of sufficiently high quantity and quality. The creative approaches to the rating task presentation and crowdsourcing program design drew upon developers' local knowledge of their own social networks, who also reported interest in the underlying problem that the data collection addresses. We consider the lessons that may be drawn from this project for implementing future iterations of the community-driven approach.

Keywords: crowdsourcing, data diversity, sentiment annotation, tools and platforms

1. Introduction

Labeled datasets for machine learning algorithms can contribute to robust and accurate models. Crowdsourcing presents one way to obtain more representative labeled datasets, but it is difficult to recruit crowd contributors from a diverse cross-section of communities. Furthermore, transactional crowdsourcing frameworks, where crowd contributors are remote performers of an annotation task designed solely by the modelers of the problem, do not fully benefit from the potential insights of the crowd. In this paper, we present an approach for involving developer communities in both annotation task development and crowdsourcing program design, so that they engage their local communities to build labeled datasets that represent their world.

Developers rooted in local communities are ideal interlopers to create tools that are effective for their particular context, as they can employ their community insights to build labeled datasets used to inform models that exemplify their locale and its people. While doing this, they can also contribute to a research space to which they might not normally have access, and as they learn, we learn with them.

For the implementation of this approach, we chose a problem in the sentiment annotation domain, where diverse judgments representative of several communities are valuable due to the subjective nature of evaluating sentiment in a given context. Specifically, we chose to collect judgments on the toxicity of Wikipedia discussion comments, where a "toxic" comment is defined as any kind of hateful, aggressive, or disrespectful comment that is likely to make someone leave a discussion. The Conversation AI team at Jigsaw, a part of Alphabet, works on technology to promote civility in online discourse. They have shown that a classifier trained using data labeled through crowdsourcing can be as effective in identifying personal attacks as the aggregate work of three contributors (Wulczyn et al., 2017).

Toxicity judgment is well-suited for the community-driven approach because toxic language is one component of the broader global problem of online harassment. According to the Pew Research Center, 41% of Americans have been the targets of harassment online, from offensive name-calling to physical threats (Duggan, 2017). In a survey commissioned by Amnesty International of women in eight countries (Denmark, Italy, New Zealand, Poland, Spain, Sweden, the United Kingdom, and the United States), 23% reported that they had been harassed or abused online, and of these, 41% felt that their physical safety was threatened (Magill, 2017). In China, over 56% of students polled at 1,438 secondary schools reported that they had been a target of online bullying (Zhou et al., 2013).

With the goal of harnessing community insights around this problem, we partnered with the Conversation AI team to test the viability of a crowdsourcing approach where both the application development and the data collection are external to the company. The expected benefits of this approach include: distributing and diversifying the sources of labeled data for a given annotation task; motivating developers and their communities to become participants in the ongoing creation of data resources; and replacing one-off tools development and maintenance with developers who are invested directly in the creative process.

2. Background

We conducted two rounds of data collection with internal or onsite developers that informed our iteration toward the community-driven approach.

2.1. Vendor-Mediated Approach

To establish a baseline for the task, we started with an industry-standard approach of developing a task interface and contracting a vendor to execute the annotation operations. The application we developed for the toxicity rating task is hosted on a Google platform. The user interface displays the Wikipedia comment, presents a range of rating options from very toxic to very healthy, supplies a box for optional comments, and lets the contributor submit the judgment. The dataset consists of 4,500 Wikipedia comments that were annotated at a redundancy of 10, resulting

in 45,000 judgments. The contributors who provided the judgments were managed by a vendor company that was selected after completing a pilot task that demonstrated delivery of labeled data at an acceptable level of quality.

2.2. Internally Managed Approach

In preparation for the community-driven approach, we worked with contractors in Singapore who had no prior knowledge of the project. We split them into two teams and conducted a three-day boot camp on Android development and the annotation task. Each team used the rest of the week to build an Android application, following simple designs, which incorporated a display of the Wikipedia comment to be rated, buttons for rating options, and space for optional comments. The backend for storing task data and judgments was again a Google platform. This exercise demonstrated that it is feasible for moderately experienced developers to create custom crowdsourcing applications without deep background knowledge of the task and within a short time frame.

Outside volunteers from local universities spent an hour of their time rating the toxicity of comments in exchange for a Google office tour and a token of appreciation (a canvas bag). Over three days, they submitted 11,809 judgments.

3. Community-Driven Approach

The community-driven approach was conducted as a challenge event in Colombo, Sri Lanka. 17 teams submitted proposals; we selected the most promising proposals, and four teams of three developers participated in the event. We provided one day of boot camp to brief the participants on the toxicity rating task, technical implementation details, and crowdsourcing program design. The backend used to store task data and judgments was available through a public API developed by the Conversation AI team.

During the Singapore round, we had observed that crowd contributors tended to get bored with the same repetitive task for an hour. We encouraged the Sri Lanka developers to be creative in their presentation of the task, and they came up with different gaming options. Since the game concepts for the applications were more involved in this round, we gave the teams a full week to complete development, and some continued to improve their applications throughout the data collection period.

As in the previous rounds, the main task was still to rate the toxicity of Wikipedia comments, but developers were given the option to extend the task to include classification of the toxicity type: insult, identity hate, obscenity, or threat.

Developers tapped into their own local networks to recruit and motivate crowd contributors. We gave the developers a month to find contributors and collect judgments, and we paid them awards for successfully building applications and for the quantity and quality of judgments that they submitted. Teams were permitted to use their own quality control methods to decide which judgments to submit. The API interspersed golden items (i.e., items with expected ratings withheld from the developers) into the dataset, allowing us to estimate the accuracy of judgments in each team's submissions based on the percentage of golden items returned with ratings consistent with the expected ratings.

We discuss each team's Android application and crowdsourcing plan in the following sections.

3.1. Forager Application

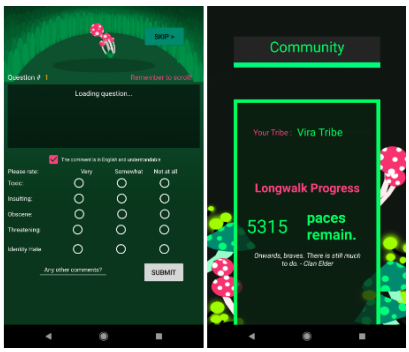

Figure 1: Screenshots from the Forager application.

In the Forager application, each user is conceptualized as a member of the same tribe on a journey. Mushrooms representing Wikipedia comments are to be rated for toxicity. In the game world, these judgments inform whether the mushrooms are safe to eat as the tribe advances along its journey. The team used word of mouth and social media platforms to advertise the application. To motivate contributors, the team offered to donate a portion of its award earnings to charity. The game appealed to contributors' sense of solidarity by placing everyone in the same tribe on a common journey. After completing a certain number of judgments, the player acquires mushrooms that count as credit for in-app purchases, designed to entice repeated engagement.

3.2. Jury Application

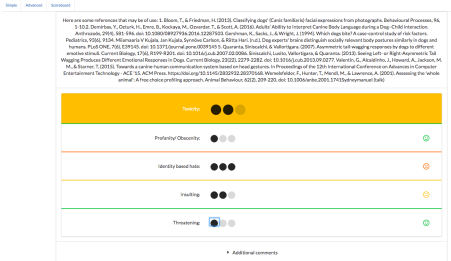

Figure 2: Screenshot from the Jury web application.

In the Jury application, users rate the toxicity of comments to earn coins. (The team's initial idea was to have users deliberate on a rating but scaled the concept down given the short development time frame.) In "simple" mode, users rate toxicity on a three-point scale; in "advanced" mode, users also classify the type of toxicity. In addition to an Android application, the team created a web application.

To recruit contributors, the team messaged friends, classmates, and colleagues about the application and shared details on social media. The application has a leaderboard and users earn coins for completing judgments. As a final motivation, the team took the three contributors with the highest number of judgments to watch the film *Justice League*.

3.3. ToxicMania Application

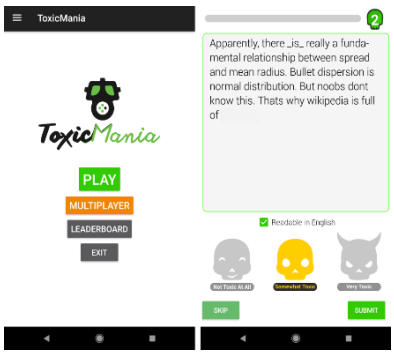

Figure 3: Screenshots from the ToxicMania application.

In single-player mode of the ToxicMania application, users compete to be at the top of the leaderboard, and they earn achievement badges as they rate comments. In multiplayer mode, users compete to outpace their friends in completing the most judgments.

The team recruited contributors through a Facebook page and through awareness sessions at their university. To motivate people to continue using the application, the team offered T-shirts to the highest-ranked players on the leaderboard. The team chose to filter out task items with lengthy comment text to reduce fatigue while playing the game.

3.4. MemifyX Application

Figure 4: Screenshots from the MemifyX application.

In the MemifyX application, users complete judgments to earn "swipes" to view or upload Internet memes. The application includes an achievements page where users earn recognition based on the judgments they have contributed. The team distributed the application to friends and sought to maintain engagement by offering the entertainment product of memes and the chance to create and view them using points. They also printed stickers to give to contributors.

4. Results

The community-driven approach of incentivizing external developers to build annotation applications and to crowdsource data collection through their own networks resulted in four different and creative solutions that brought in a high volume of labeled data at an acceptable level of accuracy, all within the time frame of a few weeks. Across all four applications, we collected 69,042 judgments with an average estimated accuracy of 76.55%. This approach was also more cost-effective than the previous approaches.

4.1. Volume

Team	Number of Judgments
Forager	23,085
Jury	22,436
ToxicMania	19,252
MemifyX	4,269

Table 1: The number of judgments submitted by each application created in the community-driven approach.

Among the four applications, Forager submitted the most judgments, while MemifyX submitted the fewest by a large margin (see table 1). One reason for this could be a waning of interest on the part of the team in promoting its application to potential contributors. It is worth noting that developers bring with them their own motivations, which may change during the course of the challenge. Including multiple teams in the challenge helped mitigate this factor. Each team's scheme for engaging contributors also likely factors into the quantity results. The Forager team motivated contributors by encouraging group togetherness and offering to donate some of the award proceeds to charity. One Forager developer said that while the points system drew in some, the application "managed to attract quite a few contributors due to our concept and artwork." ToxicMania application users reported on its Google Play Store page that it let them "contribute to something that matters while you are on commute" and "help the world while having fun." The MemifyX team, however, concentrated contributors' motivation on a scarcity of access to memes within the application. With easier ways of finding memes outside of the application, contributors may have found this motivation less appealing than contributing to a good cause.

4.2. Quality

Team	Percentage of Golden Items Correct	Golden Answers Submitted	Baseline
Forager	77.69%	3,178	57.05%
Jury	79.36%	1,857	54.53%
ToxicMania	70.84%	562	47.85%
MemifyX	81.34%	201	63.81%

Table 2: The accuracy of golden item judgments for each application in the community-driven approach.

Table 2 shows the breakdown for each application of the percentage of golden items returned with ratings matching the (withheld) expected ratings. In this case, "accuracy" refers to the percentage of golden items the team returned with ratings that matched those previously established by

the Conversation AI team. Because each team's application collects and resolves judgments from crowd contributors in a different way, we consider the ratings only in the final judgments submitted by each team. The baseline for each team indicates the expected percentage of golden items that would be returned with correct ratings by random guessing, given the distribution of golden items in that team's queue. All four teams maintained above 70% estimated accuracy, which was acceptable to the Conversation AI team and indicates that for this particular annotation task, it is possible to obtain labeled data of acceptable quality without hiring vendors, building applications internally, or finding and supervising crowd contributors.

4.3. Cost

	Vendor-Mediated	Internally Managed	Community-Driven
Number of Judgments	45,000	11,809	69,042
Cost per Judgment	$0.1168	$0.9091	$0.0628

Table 3: The number of judgments collected and cost per judgment broken down by approach.

For the vendor-mediated approach, the cost per judgment is the flat rate negotiated with the vendor. For the internally managed approach, the cost per judgment factors in payments to the onsite contractors and the cost of the canvas bag gifts for the crowd contributors. For the community-driven approach, the cost per judgment factors in: (1) a fixed award for each team that successfully developed an application; (2) a second fixed award for each team that collected at least 10,000 judgments while maintaining at least 70% estimated accuracy; and (3) a variable award for each team calculated by multiplying the number of judgments by a rate that scales with estimated accuracy.

We do not factor into the costs the company's technical investment for each approach. Externalizing annotation application development reduces Google's outlays toward recurring tools design and implementation, but it also requires ongoing maintenance of a public API.

In terms of cost per judgment, the community-driven approach is the clear winner (see table 3). This, however, is from the point of view of the payer, not controlling for other factors that may affect cost, such as the country location or quality of the data.

5. Discussion and Conclusions

Developers customized tools and incentives to fit the interests of their local community, accessing their own social networks into which we lacked insight. This community-driven approach allowed us to diversify and parallelize data collection efforts as we partnered with developer teams on different ways to solving the same problem. It afforded developers more creative control over their applications, and rather than being limited to a strict list of requirements, they could leverage their understanding of local communities and to become entrepreneurs who decided the designs and strategies that would result in the highest returns on their efforts. The approach enabled us to collect more data at a lower cost per judgment than we had previously, while still maintaining an acceptable level of overall accuracy.

The community-driven approach requires a public API and has risks, giving us less control over the development or data collection processes. Instead, it relies on the robustness of the quality control methods. It also may introduce group bias, as developers' incentive schemes could appeal to particular populations. Completing the same task in multiple communities is a possible way to mitigate this effect.

In future iterations, we plan to expand to new locations and annotation tasks. While most of the participating developers were comfortable with developing an application in a hackathon format, they were not as familiar with the basics of crowdsourcing. In order to obtain a higher quality and quantity of labeled data, we would like to provide a more in-depth training curriculum on how to create crowdsourcing programs that optimize crowd contributors' engagement and minimize their biases in judgments.

We will also consider allowing developers to choose their platform rather than requiring an Android application. The Jury team reported that potential contributors were less inclined to download an application, which prompted the team to create a web application so contributors could start rating as soon as they received the link. In Sri Lanka, 56.9% of devices used to connect to the Internet in the first half of 2017 were smartphones, while 38.1% were desktops or laptops (Department of Census and Statistics, 2017). Relaxing platform restrictions would allow developers to accommodate the usage habits of their communities and potentially expand their contributor base.

6. Acknowledgments

We thank our colleagues in Google Research and Machine Intelligence as well as Lucas Dixon, C. J. Adams, and the Conversation AI team for their work on the project's design and implementation. We also thank Keshan Sodimana for his help with the Sri Lanka event. We are grateful to the developers and crowd contributors in Singapore and Sri Lanka whose participation made this project possible. We also thank our LREC reviewers for their insights.

7. Bibliographical References

Department of Census and Statistics. (2017). Computer Literacy Statistics–2017 (First six months). Technical report, Battaramulla, Sri Lanka, June.

Duggan, M. (2017). Online harassment 2017. Technical report, Pew Research Center, 11 July.

Magill, T. (2017). Online abuse and harassment. Technical report, Ipsos MORI on behalf of Amnesty International, 19 November.

Wulczyn, E., Thain, N., and Dixon, L. (2017). Ex machina: Personal attacks seen at scale. *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399.

Zhou, Z., Tang, H., Tian, Y., Wei, H., Zhang, F., and Morrison, C. M. (2013). Cyberbullying and its risk factors among Chinese high school students. *School Psychology International*, 34(6):630–647.