Two 1%s don't make a whole: Comparing simultaneous samples from Twitter's Streaming API

Kenneth Joseph, Peter M. Landwehr, and Kathleen M. Carley

Carnegie Mellon University, Pittsburgh, PA, USA
kjoseph, plandweh, kathleen.carley@cs.cmu.edu

Abstract. In the present work, we compare samples of tweets from the Twitter Streaming API that were constructed from different connections tracking the same popular keywords at the same time. We find that tweets from the Streaming API are not sampled at random; rather, on average over 96% of the tweets seen in one sample are seen in all others. Somewhat surprisingly, however, tweets found only in a subset of samples do not significantly differ from those found in all samples in terms of user popularity or tweet structure. We conclude they are likely the result of a technical artifact rather than any systematic bias.
Practically, our results show that for those wanting more data, an infinite number of Streaming API samples are necessary to collect "most" of the tweets containing a popular keyword. Additionally, it is likely that any findings from one sample from the Streaming API will hold for all samples that could have been taken.

1 Introduction

A common method for collecting data from Twitter is to provide a set of keywords representative of a current event or trend to the "Streaming API"[1]. The Streaming API provides only a portion of the tweets matching the proscribed keywords[2], but delivers these messages in near-real time.

The Streaming API provides "enough" data for most analyses. However, situations do arise where this is not the case. For example, this limit is generally assumed to be too low when the research is aimed at testing data-hungry algorithms for pattern identification [1]. Additionally, in disaster situations, first responders may try to obtain actionable information from social media sources, particularly in the first few hours after the disaster occurs. A given sample from the Streaming API may only contain peripheral chatter from the greater

[1] https://dev.twitter.com/docs/streaming-apis

[2] We find it provides on the order of 50 tweets per second, though the more common assumption is that it provides no more than 1% of the entire volume of *all* tweets sent within a particular interval.

Twitter-sphere, therefore missing the relatively small number of tweets broadcast by victims.

Recent work suggests that even where a researcher does not want more data, her sample from the Streaming API may be a biased representation of the full dataset. Morstatter et al. [2] found that popular hashtags in a single Streaming API sample of tweets related to the Arab Spring did not represent the true set of popular hashtags used in all tweets related to the Arab Spring. Consequently, datasets that have been pulled from the Streaming API and analyzed as representative of the entire collection of relevant tweets may lead to inaccurate conclusions.

The simplest solution to issues of bias and lack of data is to get the full set of tweets pertaining to a given keyword set via the "Twitter Firehose". However, Firehose access is often prohibitively expensive. Consequently, the most popular way to access Twitter data is to use the Streaming API and ignore or design around these limitations [3]. As boyd and Crawford [4, pg. 669] note, however, "[i]t is not clear what tweets are included in. . . different data streams. . . Without knowing, it is difficult for researchers to make claims about the quality of the data that they are analyzing". Our work addresses three important open questions in this area:

- *RQ1:* How different are Streaming API samples from others taken at the same time tracking the same keywords?
- *RQ2:* Can one obtain more tweets by employing multiple Streaming API samples?
- *RQ3:* If Streaming API samples are different, do the features of tweets shared across samples differ from those that are not?

To address these issues, we use a pool of five connections to the Streaming API to track the same popular keywords at the same time. We repeat this process several times with different terms to test the robustness of our findings. With respect to *RQ1*, on average over 96% of the tweets captured by any sample are captured by all samples. This differs significantly from what is expected under uniform sampling of the full set of relevant tweets, a quantity we derive. Thus, it appears that Twitter provides *nearly* the same sample to all Streaming API connections tracking the same term at the same time.

Given the magnitude of the overlap across samples, it is not surprising that with respect to *RQ2*, a practitioner would need nearly an infinite number of Streaming API samples to capture anything close to the full set of tweets on a popular topic. However, this does not rule out the possibility that the small percentage of tweets unique to each sample are somehow different from those seen by all. To this end, and with respect to *RQ3*, we compare tweets found by different subsets of our five connections. Across metrics covering user popularity and tweet structure, we find no practically interesting differences in tweets seen in different numbers of samples. Rather, the only difference observed relates to the time tweets are delivered within sub-intervals of the sampling period (described below). We take this as evidence that Streaming API samples are slightly different only because of a technical artifact in how Twitter constructs samples on the fly. Consequently, we found no reason to suspect systematic differences exist across Streaming API samples taken at the same time on the same terms.

Our results have two important practical implications. First, we show that research conducted on a single sample of the Streaming API is likely to generalize to any other Streaming API sample taken at the same time tracking the same terms. Second, we show that if one desires more data than the Streaming API provides and is not willing to pay for it, trying to obtain more tweets using more Streaming API connections on the same keyword is not a feasible solution.

2 Related Work

Geo-spatial filters, network-based sampling algorithms [5] and user-based sampling [6], amongst several other approaches, have all been used to capture data from Twitter. Interest has also increased recently in *how* to perform sampling effectively under the constraints imposed by Twitter [7, 3], leading to innovative solutions for capturing more complete data. However, it still appears that the most common method for extracting data from Twitter is simply to specify a set of manually-defined keywords to the Streaming API and capture the resulting messages. Thus, we focus on this sampling methodology here.

There are only two articles we are aware of that explicitly consider bias across samples on Twitter. In [8], the authors compare a sample from the Streaming API to one collected from the Search API[3], stating "[t]he alternative to comparing samples to the full stream of information is to compare the two available API specifications: streaming and search". This characterization of the options available for study is not, in our opinion, complete. In understanding biases that may exist across APIs, one must first understand potential biases of each API individually, as is done here.

More recently, [2] conducted a comparison of a Streaming API sample to the full data stream on a collection of tweets gathered about the Arab Spring. They found that the set of tweets returned from a Streaming API sample provided aggregate network, topic and hashtag based metrics that did not comply with those computed on the full set of tweets matching the proscribed parameters during that same time period. Our work complements their efforts by showing that findings would almost surely have extended to any sample from the Streaming API that might have been taken at that time.

3 Methodology

Answering our three research questions required that we draw multiple, simultaneous samples of tweets from the Streaming API using identical keywords as search terms. Because drawing each sample from the API required a unique

[3] https://dev.twitter.com/docs/api/1.1

Keywords	Staggered	Keywords	Staggered	Keywords	Staggered
the	no	the	yes	i	no
the, thisisanonsenseterm	no	be	no	the, i, be	no
the, {user name}	no				

Table 1. The configurations tested in our experiment. Note each is run twice. All runs are in a separate two-hour period with all five connections.

Twitter user, we obtained access to five accounts for the purposes of this experiment. We used the Twitter API as of November 15, 2013 and carried out all connections to it using Hosebird[4], Twitter's open-source library for accessing the Streaming API.

We ran all five Streaming API connections simultaneously for two hours for each of the configurations listed in Table 1. Each configuration was run twice for a total of fourteen independent sampling periods. The configurations tested include adding a non-sensical term ("thisisanonsenseterm") to each connection, adding different terms (specifically, the name of the Twitter account for the given connection) to each connection, using multiple keywords as opposed to just one and staggering the starting time of the different connections by .15 seconds to alter sampling intervals. While one would never be interested in the keyword sets shown in Table 1, ongoing work on disasters suggests samples taken here are of comparable size to collections of disaster-related tweets collected in the few hours after a natural disaster.

After generating our samples and before analysis, we checked to ensure that there were no disconnections from the API during sampling (occasionally, Twitter will disconnect users from the API and make them re-connect). This information, along with all code and public data for the present work, can be found at https://github.com/kennyjoseph/sbp_14.

4 Results

T	The set of all tweets in the interval *matching any keyword being tracked*
C	The set of samples of T obtained via our Streaming API connections
$L_{t,c}$	Random variable denoting the likelihood that a tweet $t \in T$ is in any $c \in C$
X_t	Random variable denoting the number of samples in C in which we see tweet t
n	The size of each sample in C

Table 2. Variables used in this section.

Table 2 shows the variables we consider throughout this section- note variables are defined with respect to a single run configuration. Note that we assume n is the same for each sample in C. As $\sigma_n \approx 7$, this assumption is plausible for the derivations that follow. Across thirteen of the fourteen configurations run, n was almost exactly 367K. The one exception was one of the two configurations using the term "be", where we captured only around 335K tweets. This presumably occurred because the sample was taken early in the morning (EST), when the volume of English tweets was low.

Values of $\frac{n}{|T|}$, the proportion of the size of c to the total set of relevant tweets, ranged from .06-.30 across all runs except for the outlier noted above,

[4] https://github.com/twitter/hbc

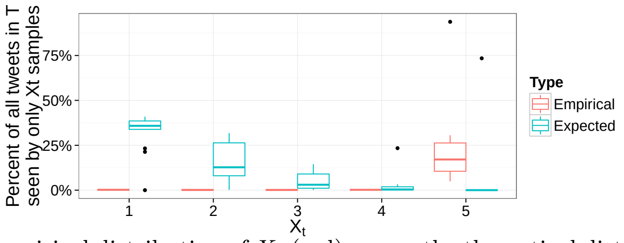

Fig. 1. The empirical distribution of X_t (red) versus the theoretical distribution (cyan) presented as box plots of values for the different configurations.

where the value raises to .94. Note that we are able to approximate the size of T by making use of the "limit notice"[5] from the Streaming API, a property not available in previous studies [2, 8, 3] and one that consequently warrants mention. Approximately once per second, the Streaming API sends a limit notice (instead of a tweet) detailing how many tweets matching the given keywords have been skipped because of rate limiting since the connection began.

4.1 RQ1: How different are Streaming API samples from others taken at the same time?

One way to understand how different samples are from each other is to compare to a theoretical baseline of how different we would expect them to be if all c provided an independent, uniform sample of T. Under this assumption, it follows that $\forall\, t, c,\ L_{t,c} \sim Bernoulli(\frac{n}{|T|})$. That is, the likelihood we get any tweet in T in a given c is equal to the number of tweets in the sample divided by the number of all tweets matching our keywords. Since $X_t = \sum_{c \in C} L_{t,c}$, we can say that $\forall\, t,\ X_t \sim Binomial(|C|, \frac{n}{|T|})$. For all of the values of $\frac{n}{|T|}$ except for our one outlier, we would thus expect the probability of X_t to decrease rapidly as X_t increases.

Figure 1 shows the distribution governing X_t as two sets of box plots; note that $X_t = 0$, the likelihood that no c observes t, is not shown. Cyan-colored box plots depict the range of possible theoretical values taken on across run configurations, while red denotes the same information for the empirical data. Black dots represent outlier data. The theoretical values for each configuration are drawn from the binomial distribution where the second parameter is set to $\frac{|C|}{5|T|}$, the mean size of the five samples.

As is clear, the empirical distribution is not binomial. Over 96% of the tweets found by any c were found by all c, something that would occur on average less than .000001% of the time if sampling were to occur randomly. Even in the outlier case, where $n \approx |T|$ and we would thus expect "most" tweets to be seen by all samples, odds of this were still significantly higher in the real data than would be theoretically expected under uniform sampling. This shows the value in comparing to a theoretically derived result and perhaps best of all indicates

[5] https://dev.twitter.com/docs/streaming-apis/messages#Limit_notices_limit

that one should expect samples taken at the same time on the same terms to be nearly identical, regardless of the size of T.

4.2 RQ2: Can one obtain more tweets simply by employing multiple Streaming API samples?

Given that Twitter makes money selling their data, it is not particularly surprising that Streaming API samples are almost identical. Again, a theoretical comparison is of use to prove this point. Equation 1 below derives $\mathbb{E}[\frac{\sum_{t \in T} I_{\neq 0}(X_t)}{|T|}]$, the expected proportion of tweets in T that we will get with $|C|$ samples, under the assumption of random sampling. Note that in this quantity, $I_{\neq 0}(X_t)$ is the indicator function that is 1 if and only if $X_t > 0$ and is 0 otherwise. Line 1 of Equation 1 uses the law of iterated expectations. Line 2 uses the fact that $\mathbb{E}[I_{\neq 0}(X_i)] = P(X_i \neq 0)$. Line 3 substitutes in the value for $P(X_t = 0)$, which equals the odds that a single c does not see a given tweet raised to the $|C|^{th}$ power.

$$\begin{aligned}\mathbb{E}[\frac{\sum_{t \in T} I_{\neq 0}(X_t)}{|T|}] &= \frac{\sum_{t \in T} \mathbb{E}[I_{\neq 0}(X_t)]}{|T|} \\ &= \frac{\sum_{t \in T} P(X_t \neq 0)}{|T|} = \frac{\sum_{t \in T} 1 - P(X_t = 0)}{|T|} \\ &= \frac{\sum_{t \in T}(1 - (1 - \frac{n}{|T|})^{|C|})}{|T|} = 1 - (1 - \frac{n}{|T|})^{|C|} \qquad (1)\end{aligned}$$

Using the result of Equation 1, we can compute that under random sampling we would need only 12 connections to capture more than 95% of the full stream when $\frac{n}{|T|} \approx .23$, the average across all configurations. Alternatively, we can use empirical data to find $n + |T| * \frac{\mathbb{E}[P(X_t = 1)]}{|C|}$, the expected number of unique tweets we will get when $|C| > 1$. Using this empirical estimate, one million connections to the Streaming API would provide us with only approximately 25% of the full data. Thus, the answer to *RQ2* for all practical situations is simply “no”.

4.3 RQ3: Do the features of tweets shared across samples differ from those that are not?

RQ3 asked how tweets unique to a subset of C differ from those observed in all samples in C. While there are only a limited number of these tweets, systematic differences between them and tweets seen in all samples could still bias analyses. Figure 2a shows summary statistics with 95% confidence intervals for the number of hashtags, URLs and mentions per tweet and the logarithm of follower and followee counts of users for tweets having different values of X_t (across all runs). While differences are statistically significant, they are so small in magnitude in each metric that they are not of practical interest. In addition, values show no obvious pattern across X_t for any metric that would indicate a systematic bias in how Twitter selects tweets to be sent with certain regularities.

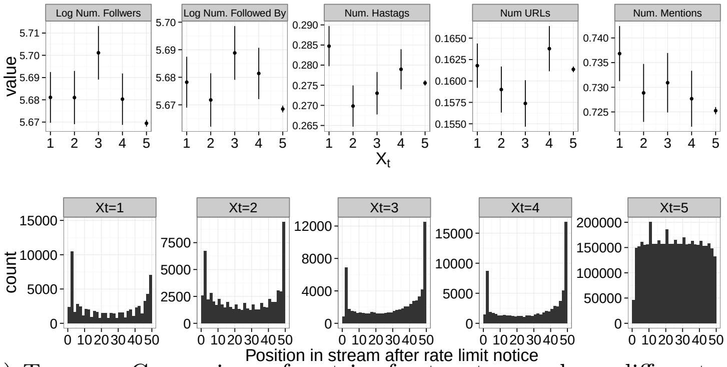

Fig. 2. a) Top row; Comparison of metrics for tweets seen by a different number of connections (each metric is a different plot). 95% confidence intervals are given using the standard error and assuming normality b) Bottom row; histogram for the position metric for all tweets with a given value of X_t. Each value of X_t is a different plot

Further support for the lack of a systematic bias that would affect analyses comes from Figure 2b, which plots histograms of the "position" metric calculated for each tweet for each value of X_t. The position metric specifies the number of tweets after a limit notice before each tweet is seen. Thus, for example, tweets with a position metric of 1 were the first tweet to be seen after a limit notice in a particular connection. Figure 2b shows the position metric for the range 0-50, which encompasses 96% of the tweets received. As we can see, tweets seen by all samples are almost equally likely to be seen any time after a limit notice. In contrast, tweets seen by a subset of C are disproportionately likely to be seen right before or after a limit notice.

This observation holds when ignoring tweets sent in the first and last few minutes of the overall sampling period, showing that the observation is unrelated to start-up or shutdown time differences across streams. While outside of the range of 0-50, distributions for the metric are much more similar across values of X_t, differences within this range suggest that tweets seen by a particular subset of C may simply be a technical artifact in how Twitter constructs samples for the Streaming API between rate limit notices.

5 Conclusion

The present work shows that Streaming API samples are not a uniform sample of all relevant tweets- rather, Twitter's technological infrastructure includes the capacity to send all connections tracking the same keywords approximately the same result. Because of this, using a larger number of Streaming API connections to track a particular keyword will not significantly increase the number of tweets collected. Other sampling methodologies, like user-based approaches [9, 6], may therefore be vital if one is to capture an increased number of tweets without resorting to purchasing data. Unfortunately, future work is needed to better understand what biases these approaches themselves introduce, and how much more data they really allow one to obtain. We also show that the few tweets unique to particular samples from the Streaming API are not different at a practical level from those seen in all samples. This holds across metrics associated with user popularity and several measures of tweet structure.

Differences between Streaming API samples consequently appear to be both slight and uninteresting. However, our work should not be taken as an indication that the Streaming API is in and of itself a random sample- it is entirely possible that Twitter holds out tweets from *all* Streaming API samples. Our findings are also restricted in that we run all samples on a single IP in a single location, use a very particular set of keywords and do not test other features of the Streaming API, such as searching via bounding boxes. While Twitter has stated that IP restrictions are not used and we expect our work to extend to other approaches to using the Streaming API, future work is still needed.

Regardless, efforts here and those we have built on lead the way to interesting future work on providing error bars for data from Twitter, for example on network metrics and the number of "needles in a haystack" we are likely to miss in disaster scenarios. As Twitter is more likely to further restrict their data than to provide more of it for free, such research is critical in our understanding of findings resulting from this increasingly popular social media site [1].

References

1. National Research Council: Frontiers in Massive Data Analysis. The National Academies Press (2013)
2. Morstatter, F., Pfeffer, J., Liu, H., Carley, K.M.: Is the sample good enough? comparing data from twitters streaming API with twitters firehose. In: The 7th International Conference on Weblogs and Social Media (ICWSM-13), Boston, MA. (2013)
3. Li, R., Wang, S., Chen-Chuan, K.: Towards social data platform: Automatic topic-focused monitor for twitter stream. Proceedings of the VLDB Endowment **6**(14) (2013)
4. Boyd, D., Crawford, K.: Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. Information, Communication & Society **15**(5) (2012) 662–679
5. Wu, S., Hofman, J.M., Mason, W.A., Watts, D.J.: Who says what to whom on twitter. In: Proceedings of the 20th international conference on World wide web. WWW '11, New York, NY, USA, ACM (2011) 705–714
6. Vieweg, S., Hughes, A.L., Starbird, K., Palen, L.: Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the 28th international conference on Human factors in computing systems. CHI '10, New York, NY, USA, ACM (2010) 1079–1088
7. Ghosh, S., Zafar, M.B., Bhattacharya, P., Sharma, N., Ganguly, N., Gummadi, K.P.: On sampling the wisdom of crowds: Random vs. expert sampling of the twitter stream. CIKM (2013)
8. Gonzlez-Bailn, S., Wang, N., Rivero, A., Borge-Holthoefer, J., Moreno, Y.: Assessing the bias in communication networks sampled from twitter. Available at SSRN (2012)
9. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the fourth ACM international conference on Web search and data mining. WSDM '11, New York, NY, USA, ACM (2011) 65–74