Twitter Streaming API Data Collection for Infrequent Keywords

T. M. Truta[1], P. Kain[1], T. Atnafu[1], A. Campan[1], J. Nolan[2] and A. Appelman[3]
[1]Department of Computer Science, Northern Kentucky University, Highland Heights, KY, U.S.A.
[2]Department of Mathematics, Northern Kentucky University, Highland Heights, KY, U.S.A.
[3]Department of Communication, Northern Kentucky University, Highland Heights, KY, U.S.A.

Abstract - *In this paper, we study the reliability of Twitter data collection when Streaming API with filtering is used. Our focus is to determine in which scenarios all tweets that contain a given keyword can be collected via the provided API interface. For this, we designed a series of experiments that use the free version of Streaming API. Our experiments show that for keywords that are infrequent (and even keywords that are moderately frequent) the entire matching tweets set can be successfully collected. We were able to determine that the Free Streaming API used with filtering provides up to 50 tweets per second and reports the number of discarded tweets in the limit entry in a cumulative way. In addition, we tested and confirmed that different types of tweets —such as normal, retweet, quote, and reply tweets — are provided by the Free Streaming API.*

Keywords: Twitter; Streaming API; data collection; sampling

1 Introduction and Motivation

It is hard to remember a time before online social networks, but it was not actually that long ago. Facebook started only in 2003 with its initial name FaceMash (renamed in early 2004), and Twitter followed in early 2006. Their adoption increased exponentially, and, as of the fourth quarter of 2018, Facebook had 2.32 billion monthly active users [14]. Twitter is significantly smaller, having "only" 321 million users during the same period [14]. Both Facebook and Twitter collect a large quantity of data from their users, and they monetize this data in various ways (via providing targeting advertising, selling data to partner companies, etc.). It is important to note that Facebook users usually share data with a "small" group of friends, and, therefore, they have an expectation of privacy for their posted data. On the other hand, Twitter users usually want to disseminate the information to a wider audience, thus their privacy concerns are small(er). Based on this, Facebook guards the privacy of its users, while Twitter does not have such privacy concerns. This makes data collection from Facebook very difficult and, more recently, it is also against the Facebook Terms of Service policy [2]. However, illegal scraping of Facebook data happened on a very large scale as recently as 2017, when Cambridge Analytica collected personal information of at least 87 million people, most of them from United States [9]. To address this unauthorized collection, Facebook increased the restriction of data access even further [13]. Twitter, on the other hand, allows tweets to be collected for research purposes and also offers APIs (application programming interfaces) that make such collection easier [6].

Twitter provides two APIs (Streaming API and Search API) that allow programmers to collect tweets in a JSON (JavaScript Object Notation) format [21]. They offer several levels of access to the APIs' features based on the payment level. Search API has three levels of access: standard/free, premium, and enterprise [22]. Its main purpose is to access historical tweets and for the standard option provides "*a sampling of recent Tweets published in the past 7 days*" [20]. The documentation lacks any information of how such sampling is obtained. Streaming API allows developers to collect tweets in real time, and it provides two access levels: Free and Decahose. The Free Streaming API returns a simple random sample of all public tweets [23]. The paid Decahose Streaming API delivers a random sample of about 10% of all current tweets [23]. Twitter Streaming API has the option of being used with specific filtering keywords; in other words, tweets that contain only specific search keywords will be collected. In the Twitter documentation there is no clear indication of how this filtering is made and if all the matching tweets will be selected.

In this paper, our primary goal is to study the reliability of Twitter data collection when Free Streaming API with filtering is used for infrequent and moderately frequent keywords. For this, we designed a series of experiments that use the free version of Streaming API. This work extends the work from [1] in which the focus was on studying popular events. A short description of this paper is presented in the next section. More specifically the contributions of this paper are as follows:

- We test whether the Free Streaming API with filtering mechanism collects tweets that we tweet from our test accounts. We concluded that all the tweets for infrequent keyword search terms (or less popular events as defined in [1]) are indeed collected.
- We aim to determine more precisely when tweets are lost by the collection method; in other words, we try to find out if the rate reported in the previous paper [1] of 600 tweets per minute is accurate. Our experiments showed that up to 50 tweets could be collected each second for very popular events that have 50 or more tweets created each second (for

a total of 3,000 tweets per minute). Due to the variation in tweets' creation times, in order to avoid losing tweets we estimate that keywords matching approximately 1,690 tweets per minute can be collected without losing any tweet, which is almost 3 times larger than the value previously reported. Our results also show that likely the number of tweets provided by the Free Twitter API with sampling has increased since 2017. In a previous work referenced in the next section [18], a significant number of tweets (over 20%) are lost for a generation rate of 1,211 tweets per second.

The remaining of this paper is structured as follows: Section 2 describes related work, Section 3 presents our experiments' design and the results of our experiments, and Section 4 presents our conclusions and suggestions for future work.

2 Related Work

Due to the ever-changing nature of Twitter APIs, it is likely that the experiments performed a few years ago will not return the same results if repeated today. Nevertheless, it is important to understand the results of related previous work in order to draw more accurate conclusions for our experiments.

In [7], Twitter data were collected during December 14, 2011 and January 10, 2012. The data collection procedure was performed via Twitter Firehose and Streaming API. Twitter Firehose represents all public tweets, and now it is replaced by the Decahose option, which represents only a tenth of the entire Firehose. In both cases, data were collected for the same set of hashtags and geographical location related to events from Syria. During this time over 1,280,344 tweets were collected via Firehose method and 528,592 tweets were collected via Streaming API. It is also interesting to note that the number of Streaming API tweets were fewer than the number of Firehose API tweets on all days (see Figure 1). At that time, Twitter API documentation allowed access to the entire set of tweets (called Firehose). This option seems to be not available anymore since November 14, 2017 [16] unless used in combination with filtering (now called PowerTrack API). In addition, at the time of collecting these data, the Twitter Streaming API could collect at most 1% of all the Tweets produced by Twitter [7]. According to Twitter usage statistics, the number of tweets per day at the end of 2011 was likely around 250 million. Even if we assume that for the winter break the number of tweets was significantly lower still this number is definitely larger than 25 million (10% of the number of tweets according to Twitter), in other words 1% of all tweets is more than 250,000 tweets per day. As you can see from Figure 1, in any day the number of tweets collected with either method is significantly lower than that. Now, based on these results, it seems that in all cases Streaming API will return just a sample of all data regardless of the data volume. There seems to be no correlation between all tweets that match that criteria and tweets collected with Streaming API. More importantly, regardless of the volume of data, Streaming API used to provide only a sample of the tweets that match that criteria.

In [5], the authors tried to extend the work from [7] by comparing samples of filtered tweets from the Twitter Streaming API that were created with five different connections tracking the same popular keywords at the same time. They find that the filtered tweets were not randomly sampled for each connection, and the samples were almost identical with an overlap greater than 96% in all cases. Their conclusion is that one cannot create a complete set of tweets for a popular keyword using this method.

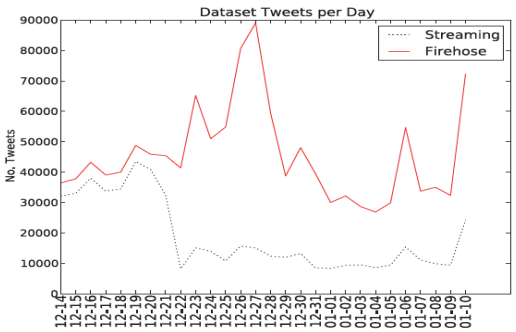

Fig. 1. Raw Tweet Counts for Experiments from [7].

In [18], Twitter data were collected during 4 different events in the first half of 2017. The data were collected using filtering with specific terms via 3 different approaches: PowerTrack, Streaming, and Search. A summary of data collection methods (we exclude Twitter Search API results since we do not analyze this API in this paper) and the number of tweets is presented in Table I.

TABLE I. NUMBER OF TWEETS CAPTURED

Time (m)	Term	PowerTrack	Streaming	Steaming /m
720	@realdonaldtrump	1,285,922	872,458	1,211
450	#ahca	122,869	122,935	273
90	#fomc	1,061	1,067	11
180	#jointsession	569,015	368,146	2,045

Data collected in Table I contradict the results from the previous paper since the number of Tweets collected via Streaming is almost the same as the number of Tweets collected via PowerTrack for two experiments. This shows that, for terms that represents a relatively small sample, the filtering results obtained with Streaming API are very similar to the ones obtained with the paid version (PowerTrack API). However, for more popular terms (*@realdonaldtrump* and *#jointsession*) this is not the case, and only ~66% of the total Tweets are retrieved with Streaming API. We noticed that the Streaming rate was quite different between the two terms (1,211 vs 2,045) and this may be because the filtering with Streaming API does not have an absolute limit but one that is dependent of other factors (such as total number of tweets and/or total number of tweets containing the search term).

Comparing the above papers published prior to 2015 [5, 7] and the last one, published in 2017 [18], it is likely that Twitter changed significantly their Streaming API with filtering, and when the number of tweets matching a keyword is not very high, the new Streaming API aims to deliver all such tweets, as for *#ahca* and *#fomc* terms.

In all three papers referenced above [5, 7, 18], the authors compare the number of filtered tweets between Streaming API and PowerTrack API, and they do not investigate how many tweets are reported during the same time by the unfiltered Streaming API. This is investigated in [1] as shown below.

Campan et al. [1] created a series of experiments in which they studied the reliability of Twitter data collection when Free Streaming API with filtering is used. The experiments utilized popular search terms (soccer terms during FIFA World Cup 2018), and they concluded the following [1]:

- For keywords that are not very popular (less than 600 Tweets per minute), it is likely that all the Tweets will be collected via Free Streaming API with filtering collection method.
- Once events are popular, some Tweets are lost. This is also reported in the Twitter log as "limit entries," and these "limit entries" seem to report the number of lost Tweets. The process Twitter is using to eliminate Tweets (with or without filtering) is likely deterministic and guided by a limit size. Such a sample cannot be considered a random sample.
- There was a limit in collecting filtered Tweets of 2,900 Tweets per minute. Such a limit did not seem to exist when collecting unfiltered Tweets -- in several instances over 5,000 Tweets per minute were collected.

However, in [1] the authors do not test their collection script via creating publically available tweets (of various types, such as normal tweets, retweets, and replies to existing tweets) that match the filtering keyword. In this paper, we perform several experiments, which illustrate the outcome when such tweets are captured by the collection mechanism. In addition, we also perform a series of experiments to determine if there is a threshold of how many tweets can be captured per minute without having tweets lost by the collection process.

3 Twitter Experiments and Results

To perform our experiments, we created several Python scripts that perform both collection and analysis of the tweets obtained using the Free Streaming API.

The Twitter Streaming API delivers tweets in a JSON (JavaScript Object Notation) format [21]. The collected tweets represent a sample of the entire Firehose if the collection script is executed without any filtering terms. Alternatively, when a search keyword is used, the Twitter Streaming API delivers only tweets that match the specified keyword. The keyword must be found in the body of a tweet, the body of quoted tweets, URLs, hashtags, and @mentions in order for the tweet to be collected [18]. The official documentation does not provide details regarding whether the Twitter Streaming API will return all such matching tweets or whether some of them are discarded.

We used Python 3.7 for our code development. In addition to Python, we used several libraries, namely *Tweepy* [19], *Pandas* [10], *NumPy* [8], and *RE* [12]. All experiments were performed on identical virtual machines (we had 3 available) running Windows 2012 Server with 4 GB RAM hosted on a Dell private cloud. For authentication to the Twitter Streaming API, we used five Twitter user accounts. For creating tweets during our collection windows, we used four Twitter accounts.

While running our collection script, we noticed that we have potential time synchronization issues between the Twitter time and our virtual machine times. Twitter is providing the creation time of each tweet in Coordinated Universal Time (UTC), and our virtual machines are using Eastern Standard Time (EST). All scripts were designed to use the same time (UTC). In addition, the creation time of the Tweet is not the same as its delivery time. For most tweets the delay is similar (our experiments showed approximately 5 seconds); however, there are tweets that are *delivered* with an additional delay of a few seconds (we found delays of up to 5 more seconds). To make sure we collect such tweets, we include a buffer of 1 minute at the start and at the end of each collection window. Additional tweets are discarded in our processing scripts based on the creation time; for all the experiments presented below, the times reported are based on tweets' creation time, not their collection time.

The design of experiments as well as their results are presented next.

Experiment 1 – *Collection of tweets using search keywords of different frequency levels.*

In this experiment, we collect tweets using several search keywords, namely "*bibbidibobbidiboo*," "*NKU*," "*Kentucky*," "*Zidane*," and "*the*." This experiment is divided in 5 collection windows of 10 minutes each. For each collection window, we run in parallel on two virtual machines the same collections script. During each collection window, we publish 10 tweets using two Twitter accounts. All these tweets contain the corresponding keyword for which we collect the related tweets. Five tweets are regular tweets, and 5 tweets are retweets of tweets that contain the targeted keyword.

For the first keyword "*bibbidibobbidiboo*" (the magic keyword used in Cinderella story by her fairy grandmother), we do not expect to collect any tweet with this keyword during our collection window except tweets that we intentionally created. For the second keyword "*NKU*" (Northern Kentucky University), we expect to collect a few additional tweets in addition to the ones we publish. For the third ("*Kentucky*") and fourth ("*Zidane*") keywords, we expect to collect a larger number of tweets. While for the "*Kentucky*" keyword, we expect to collect all 10 tweets, for "*Zidane*" keyword there is a possibility that there will be a large number of tweets and that some of them may not be collected. This is also because we intentionally selected one of the highest trending Twitter topics according to trends24 website [17] on the day when we conducted this experiment (Zinedine Zidane, the former French soccer player and coach of Real Madrid, was selected as the new Real Madrid coach). For the last keyword ("*the*"), we expect many tweets will not be collected since the number of matching tweets will likely be above the limit imposed by Twitter; thus, we expect to collect only a sample of the 10 tweets we publish with this keyword. The word "*the*" was the most common word across all the tweets according to [4], and

it is likely very common at any given collection time. The results of this experiment, as collected on the first virtual machine, are shown in Table II. Please note that the Coordinated Universal Time (UTC) (the time shown in the Table II) is 4 hours ahead of the Eastern Standard Time (EST) (the local time of the experiments). As expected for the first three keywords, we captured all the 10 tweets we created during the collection window because those words are infrequently tweeted. For the last two keywords we were able to collect only 6 (for "*Zidane*"), and 0 (for "*the*") of our original tweets, since these two keywords were sampled in some way due to the high frequency of their use on Twitter.

TABLE II. EXPERIMENT 1 RESULTS

	bibbidibo -bbidiboo	NKU	Kentucky	Zidane	the
Start Time (UTC)	03-11-19 16:46:00	03-11-19 17:04:00	03-11-19 17:18:00	03-11-19 17:32:00	03-11-19 17:45:00
Duration (min)	10	10	10	10	10
# Created Tweets	10	10	10	10	10
# Collected Tweets	10	12	240	29,941	30,003
# Created Tweets that are Collected	10	10	10	6	0

The concurrent collection of tweets on both virtual machines produced identical results on the first three keywords and slightly different results for the last two keywords. We summarize the results on both virtual machines for the last two keywords in Table III.

TABLE III. EXPERIMENT 1 RESULTS - "ZIDANE" AND "THE" KEYWORDS

	Zidane(VM1)	Zidane(VM2)	the(VM1)	the(VM2)
Start Time (UTC)	03-11-19 17:32:00	03-11-19 17:32:00	03-11-19 17:45:00	03-11-19 17:45:00
Duration (min)	10	10	10	10
# Collected Tweets	29,941	29,938	30,003	30,014
	Zidane(VM1) ∩ Zidane(VM2)		the(VM1) ∩ the(VM2)	
Union (∩)	29958		30134	
	Zidane(VM1) ∪ Zidane(VM2)		the(VM1) ∪ the(VM2)	
Intersection (∪)	29921		29883	
	Zidane(VM1) \ Zidane(VM2)	Zidane(VM2) \ Zidane(VM1)	the(VM1) \ the(VM2)	the(VM2) \ the(VM1)
Minus (\ )	20	17	120	131
	Zidane(VM1)	Zidane(VM2)	the(VM1)	the(VM2)
Tweets Not Collected	~ 13,247	~ 13,249	~ 336,390	~ 336,377
Total Tweets	~ 43,188	~ 43,187	~ 366,393	~ 366,391
% Tweets Not Collected	~ 36.67%	~ 36.67%	~ 91.81%	~ 91.81%

For tweets not collected by our scripts, we looked at the values reported by the limit entries. These limit entries record in the "collected tweets" file the number of tweets that should have been collected but were discarded due to the high volume [1]. However, between two limits entries that are recorded just before and after our start and end time, it is difficult to know if the lost tweets happened in our collection windows or just before or after. In our reported numbers of lost tweets for this experiment, we considered the last limit entry value before the start of the collection window time (in both cases this was in the second proceeding the start of the collection window) and the last limit entry within the collection windows time. Therefore, the reported numbers, for the last three columns, only approximate the correct values, thus the slight variation (of only two tweets) in the number of total tweets.

From Table III and the entire experiment, we observe the following:

- The concurrent collection of tweets produces almost identical results with very few tweets that are captured on only one virtual machine. This is a similar result as in [1].
- Our experiment also supports the conclusion that the limit entry reports the number of the discarded tweets as in [1].
- When collecting the tweets that match keyword "*the*", we noticed that there is one limit entry after exactly 50 tweets. This pattern repeats until the end of the collection. Moreover, 50 tweets is the number of tweets that were delivered in a second, and we can, in general, notice that the second for the tweet creation changes with every new limit entry. We also observed that we get very close to 30,000 tweets for this experiment (which means 50 tweets per second). Looking closely at individual tweets, we noticed that the creation time of the tweet and the delivery time differs, and we occasionally have older tweets delivered after newer tweets. In Table II and III, we report the number of tweets based on the tweet creation time; thus, there is a slight variance (of only 3, respectively 14 tweets) compared to the delivery rate of 30,000 tweets per 10 minutes.
- When collecting the tweets that match "*Zidane*" keyword, we noticed similarities with the keyword "*the*". There are many times when between two limit entries exactly 50 tweets are delivered; however, there are situations when there are more than 50 tweets delivered. In these situations, the limit entry timestamps show that there is more than a second between such entries. This is likely because for a small period Twitter can deliver all matching tweets; in that case, there are no lost tweets to be reported. Based on this, we can conclude that Twitter provides no more than one limit entry per second and delivers a constant rate of 50 tweets per second if enough tweets are matching the specified keyword.

Experiment 2 – *Collection of different types of tweets.*

The search term for this experiment is "*bibbidibobbidiboo*". During the 10 minutes collection window the following tweets were published from four accounts (labeled A1 and A2):

a) Publish the tweet: "Tweet 1: #*bibbidibobbidiboo*" (A1)
b) Retweet the above tweet, no text addition (A2)
c) Publish the tweet: "Tweet 2: *bibbidibobbidiboo*" (A1)
d) Retweet the above tweet, with text addition "Retweet" (A2)

e) Publish the tweet: "Tweet 3: *bibbidibobbidiboo*" (A1)
f) Retweet the above tweet, with text addition "Retweet *bibbidibobbidiboo*" (A2)
g) Publish the tweet: "Reply" (A1)
h) *Reply to the above tweet with text "Reply" (A2)*
i) Publish the tweet: "Tweet 5: *bibbidibobbidiboo*" (A1)
j) Reply to the above tweet with text "Reply *bibbidibobbidiboo*" (A2)
k) Publish the mentions tweet: "@A2 Tweet 6: *bibbidibobbidiboo*" (A1)
l) Retweet the above tweet, no text addition (A2)
m) Publish the tweet: "Tweet 7: *bibbidibobbidiboo*" (A1)
n) Retweet the above tweet, with text addition "Retweet 1" (A2)
o) *Retweet the above retweet, with text addition "Retweet 2" (A1)*

Please note that the retweet with text addition is also known as a *quote* tweet.

All above tweets except the tweets created at Steps h) and o) were collected by the collection script. The reply tweet (Step h; shown in Fig. 2) is not collected since it does not contain the original tweet nor the search term. This reply tweet is visible to the A1 (*@trutat1* in Fig. 2) account and to A1 followers. It is not visible to A2 followers (*@TweetsResearch*). Such a reply will not increase the visibility of the original tweet.

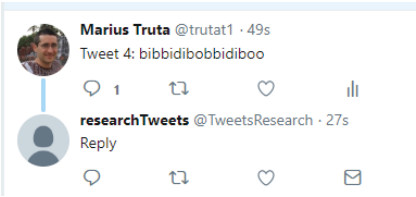

Fig. 2. Reply Tweet.

The retweet to the retweet with text addition that do not contain the search keyword (Step o; shown in Fig. 3) is not collected since the original tweet is replaced by a hyperlink and is no longer included in the retweet in the original form. Unlike the reply to a tweet, the retweet with comments is available to all followers to the account that performed the retweet. In addition, it is worth noting that a retweet with comment is not counted in the retweet category.

The above differences between a retweet and a reply to a tweet show that in a rigorous analysis of a tweets collection, one must consider different categories of tweets that are collected.

Fig. 3. Retweet to Retweet with Text Addition.

Since most retweets are without comment, we wanted to test what happens when we have retweets to retweets to retweets between multiple accounts. For this, we continued Experiment 2 with the created tweets listed below for a new 10-minute collection window. For this part of the experiment, we used four distinct accounts (A1, A2, A3 and A4).

p) Publish the tweet: "Retweet 1: #*bibbidibobbidiboo*" (A1)
q) Retweet the above tweet, no text addition (A2)
r) Retweet the above tweet, no text addition (A3)
s) Retweet the above tweet, no text addition (A4)

All the above four tweets were captured by our collection script; the final retweet is shown as in Fig.4, with the value of 3.

Fig. 4. Retweets whitout Text Additions.

The above experiment shows that retweets with comment (quote tweets) are quite different than retweets without comment, so they must be considered as two different types of tweets.

The final tests that we perform as part of this experiment are related to reaching the maximum tweet size (currently 280 characters). For this, we started the collection script, and we published the following tweets during the creation window:

u) Publish the tweet: "Test 1. Garbage text that has a total of 278 characters (missing text): *bibbidibobbidiboo*." (A1)
v) Retweet the above tweet, no text addition (A2)
w) Retweet the above tweet, no text addition (A3)
x) Retweet the above tweet, with text addition "Nice Tweet 1. Now over 280 characters." (A4)
y) Publish the tweet: "Test 1. Garbage text that has a total of 278 characters (missing text): *bibbidibobbidiboo*." (A1)
z) Reply to the above tweet with text "Reply *bibbidibobbidiboo*" (A2)

Our collection script captured all six tweets from above; therefore, we can conclude that replies and retweets with comment do not cause any truncation of the original tweet.

We conclude that regardless of the tweet type, if the search keyword is part of the tweet the tweet will be identified for collection by the Twitter Free Streaming API.

Experiment 3 – *Collection of tweets using more frequent search keywords.*

For our last experiment, we estimate the rate of tweets that can be captured per minute such that no tweets will be lost. For this, we performed the tweet collections shown in Table VI.

TABLE IV. EXPERIMENT 3 DATA

	Beto	Christchurch *	today
Start Time (EST)	03-14-19 14:00:00	03-15-19 8:10:00 (1 hour) 03-15-19 13:10:00 (9 hours)	03-17-19 23:30:00
Duration	10 hours	10 hours (1 + 9)	10 hours
# Collected Tweets	389,667	1,206,575	900,266
# Lost Tweets	0	108,385	1348
# Total Tweets	389,667	1,314,960	901,614

* Our collection script stopped due to a network issue and we restarted it later.

We selected "*Beto*" as our first keyword since it was one of the trending Twitter topics on that day (Beto O'Rourke announced on March 14, 2019 his candidacy for the Democratic nomination for President of the United States) [17]. For the second keyword, we choose "*Christchurch*" due to the terrorist event that happened on March 15, 2019. As expected, this was a major event being by far the most talked topic on Twitter on the collection day. The last keyword is a very common word ("*today*") that we expect to be quite common in many tweets regardless of the collection window time.

In Fig. 5, we present the number of collected tweets for each individual hour from the collection window. We notice that for "*Beto*" and "*Christchurch*" the number of tweets have a descending trend, while for "*today*" there is dip in the middle of our collection. The descending trend for the first two keywords is easily explained since both were triggered by a specific event, and interest specific events tend to decrease over time. For "*today*" keyword, the timing of the collection window (11:30 PM – 9:30 AM EST) likely influences the trend. We notice that the last 2 hours are the early working hours on the US east coast.

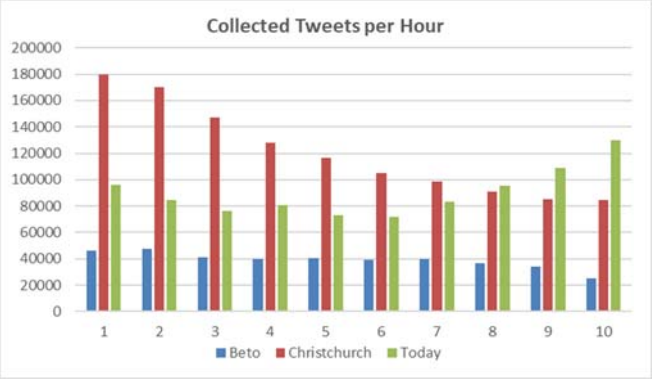

Fig. 5. Collected Tweets per Hour – Experiment 3

In Fig. 6, we present the number of lost tweets for each individual hour from the collection window. These tweets are computed based on the limit entry values as explained in the Experiment 1. The values for the first 3 hours for "*Christchurch*" collection were: 94,572, 12,022, and 1,606. There are no lost tweets for the "*Beto*" keyword.

In order to understand more with respect to the pattern of tweets for individual keywords, we look at the average and standard deviation values for all tweets (collected or total) for each minute and second. Specifically, for a collection window, we determine how many tweets are collected and lost, and their total for each individual minute (or second). However, we can easily see from Fig. 5, there is a large variation between different hours; therefore, we would like to look at how the average and standard deviation changes for each hour (when data are collected per minute) and for each minute (when data are collected per hour). We compute the moving average, moving standard deviation, and relative moving standard deviation, accordingly [3]. Table V shows the average of the moving average (`AVG(MAVG)`), moving standard deviation (`AVG(SD)`), and relative moving standard deviation (`AVG(RSD)`) for our experiments. To avoid any bias, we considered only the last 8 hours of data for "*Christchurch*" experiment (to avoid averaging data collected hours apart due to the networking failure we had during our collection).

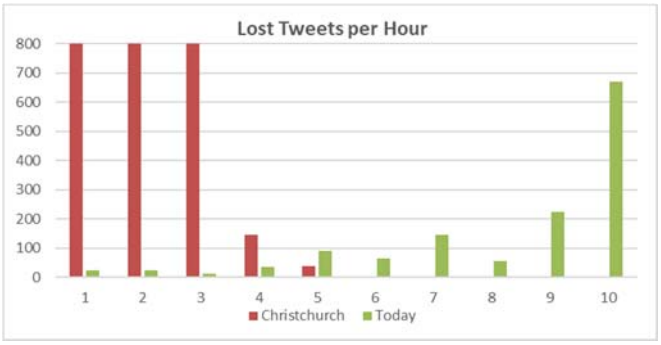

Fig. 6. Lost Tweets per Hour – Experiment 3

From Table V, we observe that the standard deviation is higher for "*today*" than for "*Christchurch*" when we collect data per minute. Such variation does not exist when data are collected per second. From this table we can also estimate what is the average number of collected tweets per second (x) such that it is very unlikely that there will be any lost tweets per second. As determined in Experiment 1, up to 50 tweets can be collected per second. Based on this, we assess that 50 will be x + 3 times the standard deviation (we use the 68–95–99.7 rule for this computation [11]). Assuming a relative standard deviation of 25.8%, our estimate is 28.18 tweets per second (1,690 tweets per minute). The relative standard deviation value of 25.8% represents the 95% percentile of the standard deviations computed for total tweets numbers ("*today*" keyword).

TABLE V. EXPERIMENT 3 - AVERAGES AND STANDARD DEVIATIONS FOR TOTAL TWEETS

		Data Collected for each second; moving average computed for each minute		Data Collected for each minute; moving average computed for each hour	
		Christchurch	today	Christchurch	today
Collected Tweets	AVG (MAVG)	33.51	24.99	1870.73	1454.35
	AVG (SD)	5.64	5.12	72.86	135.79
	AVG (RSD)	17.47%	20.78%	3.79%	9.39%
Total Tweets	AVG (MAVG)	36.50	25.03	1884.69	1456.19
	AVG (SD)	6.23	5.17	80.84	143.34
	AVG (RSD)	17.74%	20.91%	4.06%	9.87%

In Table VI, we summarize in how many intervals (with size one minute) we collected the specified range of tweets for both "*Christchurch*" and "*today*" keywords. We also include in how many such intervals we have lost tweets. We include only minutes where the range of collected tweets are between 1,400 and 2,400 because for all minutes with less than 1,400 collected tweets, no tweet was lost, and for all minutes with more than 2,400 collected tweets, each minute had lost tweets.

TABLE VI. EXPERIMENT 3 SUMMARIZATION

# Collected Tweets	Christchurch Minutes	Christchurch Minutes Lost Tweets	today Minutes	today Minutes Lost Tweets
1400 - 1499	100	0	63	1
1500 - 1599	39	0	59	0
1600 - 1699	56	0	66	1
1700 - 1799	50	0	27	3
1800 - 1899	26	0	18	3
1900 - 1999	32	5	23	7
2000 - 2099	31	11	13	3
2100 - 2199	23	12	16	13
2200 - 2299	24	18	15	11
2300 - 2399	17	16	8	8

From the above table we notice that "*today*" keyword produced several minutes with low count (2 minutes under 1,690 tweets) that had a few lost tweets, which suggest that during these minutes there is one second where the number of tweets exceeded 50. Looking closely at the data, we noticed that this happens at very specific times (for instance at 6:30:00 AM and 9:00:00 AM EST). We believe that this is because some users chose to post tweets automatically at given times and of course "*today*" is a common keyword in those automatic tweets (we found an unexpectedly large number of weather related tweets in the first second after 6:30:00 AM EST time). We did not see any such spikes for "*Christchurch*" keyword. For "*Christchurch*," the lowest count minute where tweets were lost was 1,910, which is significantly higher than our estimation of 1,690.

Overall, in this third experiment, we were able to approximate the number of tweets that can be captured in a minute without having tweets lost (of 1,690 tweets). In addition, we determine that automatic tweets tend to be set up at change of the hour (or at 30 minute past the hour) and occasionally this may lead to unexpectedly lost tweets.

4 Conclusions and Future Work

In this work, we studied the reliability of Twitter data collection when Free Streaming API with filtering is used for infrequent and moderately frequent keywords. Our experiments showed that the tweets for infrequent keyword search terms (up to ~ 1690 tweets per minute) are collected without any losses for most minutes. In addition, we determined that Twitter is providing up to 50 tweets per second and it reports the number of discarded tweets in the limit entry in a cumulative way. We also investigated how different types of tweets are captured via Streaming API. Our experiments show that collecting filtered Twitter data on matching keywords can be performed without loss of tweets for most of the keywords, and definitely for any keyword that is not very frequent. Researchers that need Twitter data for their studies can use the approach we introduced in this paper to collect data from Twitter based on their needs.

5 References

[1] A. Campan, T. Atnafu, T. M. Truta, J. Nolan, "Is Data Collection through Twitter Streaming API Useful for Academic Research?," IEEE International Conference on Big Data (Big Data), pp. 3638–3643, 2018.

[2] Facebook Terms of Service, Available at: www.facebook.com/terms.php, 2019.

[3] T. Finch, "Incremental Calculation of Weighted Mean and Variance" (PDF). University of Cambridge, Available at: http://people.ds.cam.ac.uk/fanf2/hermes/doc/antiforgery/stats.pdf, 2009.

[4] L. Grossman, "The 500 Most Frequently Used Words on Twitter," Available at: http://techland.time.com/2009/06/08/the-500-most-frequently-used-words-on-twitter, 2009.

[5] K. Joseph, P. M. Landwehr, and K. M. Carley, "Two 1%s Don't Make a Whole: Comparing Simultaneous Samples from Twitter's Streaming API," In Social Computing, Behavioral-Cultural Modeling and Prediction, Springer, pp. 75 – 83, 2014.

[6] J. Littman, "Where to Get Twitter Data for Academic Research," Available at: https://gwu-libraries.github.io/sfm-ui/posts/2017-09-14-twitter-data, 2017.

[7] F. Morstatter, J. Pfeffer, H Liu, and K. M. Carley, "Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose," Proceedings of the AAAI Conference on Weblogs and Social Media, 2013.

[8] NumPy, Available at: http://www.numpy.org, 2019.

[9] D. O'Sullivan, D. Griffin, and P. DiCarlo, "Cambridge Analytica's Facebook Data was Accessed from Russia, MP says," CNNTech, Available Online at: https://money.cnn.com/2018/07/17/technology/cambridge-analytica-data-facebook-russia/index.html, July 17, 2018.

[10] Pandas, Python Data Analysis Library, Available at: https://pandas.pydata.org/, 2019.

[11] F. Pukelsheim, "The Three Sigma Rule", American Statistician 48, 1994.

[12] RE, Regular Expressions in Python, Available at: https://docs.python.org/3/library/re.html, 2019.

[13] M. Schroephfer, "An Update on Our Plans to Restrict Data Access on Facebook," Available at: https://newsroom.fb.com/news/2018/04/restricting-data-access, April 4, 2018.

[14] Statista Facebook, Available online at: www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide, 2019.

[15] Statista Twitter, Available at: www.statista.com/statistics/274564/monthly-active-twitter-users-in-the-united-states, 2019.

[16] A. Tornes, "Introducing Twitter Premium APIs," Twitter Developper Blog, Available at: https://blog.twitter.com/developer/en_us/topics/tools/2017/introducing-twitter-premium-apis.html, 2017.

[17] Trends24, Available at: https://trends24.in, 2019.

[18] R. Tromble, A. Storz, and D. Stockmann, "We Don't Know What We Don't Know: When and How the Use of Twitter's Public APIs Biases Scientific Inference," SSRN, Available at: https://ssrn.com/abstract=3079927, 2017.

[19] Tweepy, Available at: www.tweepy.org, 2019.

[20] Twitter Developer, Available at: https://developer.twitter.com, 2019.

[21] Twitter Objects, Available at: https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json.html, 2019.

[22] Twitter Search API, Available at: https://developer.twitter.com/en/docs/tweets/search/overview, 2019.

[23] Twitter Streaming API, Available at: https://developer.twitter.com/en/docs/tweets/sample-realtime/guides/recovery-and-redundancy, 2019.