Surpassing the Limit: Keyword Clustering to Improve Twitter Sample Coverage

Justin Sampson, Fred Morstatter, Ross Maciejewski, Huan Liu
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
{justin.sampson, fred.morstatter, ross.maciejewski, huan.liu}@asu.edu

ABSTRACT

Social media services have become a prominent source of research data for both academia and corporate applications. Data from social media services is easy to obtain, highly structured, and comprises opinions from a large number of extremely diverse groups. The microblogging site, Twitter, has garnered a particularly large following from researchers by offering a high volume of data streamed in real time. Unfortunately, the methods in which Twitter selects data to disseminate through the stream are either vague or unpublished. Since Twitter maintains sole control of the sampling process, it leaves us with no knowledge of how the data that we collect for research is selected. Additionally, past research has shown that there are sources of bias present in Twitters dissemination process. Such bias introduces noise into the data that can reduce the accuracy of learning models and lead to bad inferences. In this work, we take an initial look at the efficiency of Twitter limit track as a sample population estimator. After that, we provide methods to mitigate bias by improving sample population coverage using clustering techniques.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering; Retrieval models

Keywords

Clustering; Text Processing; Social Media

1. INTRODUCTION

The use of social media as a data source has allowed for an extremely wide-reaching range of topics and phenomena to be studied on a large scale. Highly prominent among sources for data gathering is the microblogging site Twitter. The popularity of Twitter as a source can be attributed to the type of data that is produced by its users, tweets must be concise due to the 140 character limit imposed by the service, and the data gathering limits are relaxed compared to most other similar services. The combination of these favorable traits, found almost uniquely in Twitter, have allowed it to become organically selected as the "'model-organism' for research with social media data" [21]. The Twitter streaming API allows users to gather up to 1% of all tweets that pass through the service at any time. According to Twitter, around 500 million tweets are posted every day meaning that a single user can gather up to 5 million tweets in this period of time.[1] While this generous data rate allows for large samples to be gathered over time, rate limiting still poses significant challenges for any research which requires as close to a complete data set as possible.

Recent works have shown that there is significant bias in the sampling method used by Twitter's filtered streaming API. However, this sampling method remains unpublished, making it difficult for end users to detect and correct for the resulting bias in their data sets. Attempting to create generalized models or make any form of measurement based on a data set with ingrained bias is dangerous and can result in large margins of error that may not be acceptable. Additionally, the inverse is also true. When the total population size for a data set is known bias can be minimized making it possible to make good predictions based on principled statistical and machine learning techniques. In the absence of the ability to measure and correct for sources of bias, the only available recourse is to ensure that the coverage of the gathered sample is as close as possible to the total sample population during the gathering process. However, in order to measure the difference between a sample and the complete set, a useful population measure is required.

The Twitter streaming API uses a mechanism called "limit track" which informs the user of the number of tweets that were not delivered to them due to exceeding the 1% rate limit. The limit track data is provided periodically along with tweets delivered in a particular stream. Unfortunately, if the limit track value provided by Twitter is not a reliable measurement, then it becomes significantly more difficult to determine the overall sample population, and, as a result, the level of bias in the sample remains unknown. In addition, the usefulness of the limit track value is further reduced as it does not allow for any method to retroactively obtain the lost data. Unfortunately, since the method used for sampling tweets, as well as how the limit track value is obtained, is not yes published, it is imperative to know (1) whether Twitter's

HT'15, September 1–4, 2015, Guzelyurt, TRNC, Cyprus.
 ISBN 978-1-4503-3395-5/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2700171.2791030.

[1] https://about.twitter.com/company

limit track is accurate and, if it is not, we (2) must find another way to decide if information is being lost.

With these limitations in mind, this work will attempt to answer the following questions:

- Is the Twitter limit track an accurate measure of the amount of data missing from a sample?
- How can we structure our search to reduce the volume of missed data?

While data sets created using completely random sampling methods are known to preserve important statistical properties that allow a smaller sample size to generalize well to the entire set [7, 19], the introduction of deterministic processes into sample creation introduces bias that can cause erroneous and dangerous conclusions to be drawn from the data. Unfortunately, the incredible size and rigidity inherent to the infrastructure of these services, such as Twitter, do not necessarily lend themselves well to producing the type of random sampling necessary. This shifts the burden of responsibility for creating and using unbiased data from the service to the user. However, producing unbiased samples from inherently biased data sources is nontrivial. Regardless of biases in the sampling method, increasing the ratio of coverage between the sampled data and the complete dataset will reduce sample error and improve results. While gathering the complete set of data would be optimal, most services impose strict sampling rate limitations. In other cases, where the complete data samples are available, exorbitant pricing can be a roadblock. This work proposes several methods to overcome these limitations by increasing sample coverage, thereby minimizing bias in incomplete samples. Though the proposed methods were implemented and tested on Twitter, the results should generalize well to any keyword-based data gathering services.

2. RELATED WORK

The predictive power of social media services, such as Twitter, has been used to effectively track and predict the spread of disease [1, 6, 8]. Other efforts have also shown promising results by using social media to discover a wide range of collective knowledge such as real-time political polling [4, 22] and the potential success of movies at box-office [2, 14]. However, there are very few standards governing how data from social media is gathered and how research and predictions should be approached [16]. A number of studies have shown that the method used for sampling can introduce various forms of bias which introduce error into results and remain largly unnoticed [9, 19]. Very little research has gone into methods for correcting for bias. Morstatter et al. proposed using bootstrapping [5] and secondary data sources to obtain a confidence interval for the occurrence rate of a hashtag between the two sources. Such a statistical difference could be used as a red flag for the presence of bias in the stream [18].

Several relevant works have attempted to uncover the underlying mechanisms with which Twitter disseminates tweets through its various streaming APIs. In 2013, Morstatter et al. tested the differences between the public Twitter streaming API, which is commonly used by researchers but will only return up to 1% of all available data, and the prohibitively priced "firehose" stream, which offers all of the available data to those willing to pay for it [19]. This work uncovered a number of anomalies such as varying top hashtags, differing topic distributions, and varying network measures when using the same searches on both the paid and free services. They explain that according to the "law of large numbers" if the sample is truely random then it should relate well with the results from the population, in this case the "firehose" results [19]. These differences indicate that the streaming API introduces some form of bias when determining how to limit the results sent to a stream [18].

The possible causes of this sampling bias have been a continuing source of inquiry. Joseph et al. used five independent streaming sources the differences between multiple streaming results taken at similar but varying time windows [10]. They used a set of keywords and usernames including stop words such as "the" and "i" and words that they specifically invented for the experiment. In order to determine if starting time had an impact on the results they staggered each start by a fraction of a second. After running these tests multiple times, they showed that, when using the same keywords across each stream, 96% of unique tweets were captured by all streams [10]. Since the experiment used stop words that undoubtedly make up a significant portion of English tweets, a random distribution method would have given results that varied wildly across the separate streaming connections. This is strong proof that the sampling method used by Twitter is highly deterministic.

Kerg et al. found further evidence of non-random sampling. In earlier forms of the Twitter streaming API, the three data streaming levels which provided 1%, 10%, and 100% of Twitter data were named "Spritzer", "Gardenhose", and "Firehose" respectively. These streams, unlike the filter-based stream, provide data from the entire body of current tweets. Through analysis of the unique tweet IDs provided by Twitter, the authors discovered that the unique IDs included data, such as the timestamp when the tweet was created, the data center that created it, and other information related to the underlying technical infrastructure of the Twitter data centers. Analysis of the timestamp in particular showed that, in the limited streams, the timestamps only fell over a specific interval of milliseconds directly related to the percentage of sample coverage specified by the service level [11]. Though no similar timestamp-based sampling method appears to be used in the filtered search stream, the non-random nature of the filtered data, in addition to the use of simple deterministic methods in past dissemination schemes, indicates that there are underlying artifacts in the filtered stream infrastructure as well that may be adding bias to gathered samples.

The 1% boundary has proven to be a significant hindrance for applications and research that require as close to a complete set of data as possible. These include mission critical response situations, such as those necessary for emergency response and monitoring applications [12, 17, 23], as well as any form of research that is highly affected by sample bias. As a direct result of the high cost of the "firehose" service, many users have attempted to develop novel solutions for gathering data to improve either the size and coverage of the dataset [13] or the overall quality of results for a smaller sample [7]. Li et al. proposed a system that uses "automatic keyword selection" to determine the best search terms to use for a given search through a keyword cost-based system. Using this method improved topic coverage from 49% of targeted tweets obtained through human-selected keywords to 90% in the automatic system [13]. Ghosh et al. took an alternate approach and attempted to improve the quality

of their sample by gathering topic-based twitter data from experts and comparing the "diversity, timeliness, and trustworthiness" to a larger sample of community tweets of the same topic. While the expert-based search did show an improvement in all of these areas, they cautioned that crowd-based data could not be entirely discounted as it captured other significant properties such as the flow of conversation in the topic that is otherwise ignored by experts [7].

3. KEYWORD SPLITTING APPROACH

The Twitter Streaming API allows anyone to gather real-time data from Twitter by specifying a set of parameters that include search terms, user names, or location boxes. In the case that the search terms specified for a stream surpass the 1% limit, the API informs the user of the total number of tweets missed since the streaming connection began. Ideally, this would give the user a quantitative measure of the overall sample size for their search. The total size of the dataset would then be the sum of the number of unique tweets gathered added to the limit track value provided by Twitter. Knowing the exact quantity of missing data is of paramount importance when it is necessary to adjust the data gathering method to account for gaps in the sample.

The Twitter limit track is designed to give a measurement of lost data for a single stream. However, our proposed methods revolve around using multiple streams to increase the effective sample size. In order to determine if the limit track is a useful measure for the overall sample, when viewed from the context of multiple streams, we ran a number of trials simultaneously. At the first level, all keywords used in the search were tracked using a single stream. For each level beyond the first, the keywords were separated as evenly as possible among the streams. In the example shown in Figure 1, all keywords are split between crawlers based on the number of crawlers required at each level. For example, split level two separates all 400 keywords between two crawlers. All split levels are run simultaneously up to a maximum split level of five which required a total of fifteen streaming clients. After a set period of time, all crawlers terminate and any duplicate tweet IDs are discarded from the set for each split level. Since no keywords were added or duplicated between the streams, the total number of possible tweets should be equivalent to the unsplit streams number of caught tweets as well at the reported limit value. However, in nearly every experiment, there was always a number of splits that would result in a larger number of unique tweet IDs than should be possible according to limit track. As shown in Figure 1, we accumulated 107.3% of the tweets that were indicated by the limit track, meaning that we received more tweets than were estimated by Twitter. Furthermore, using a four-split approach, we collected 137.2% of the tweets indicated by the limit track In order for the limit track to be an accurate measurement of the sample population it should not be possible to gather unique tweets much beyond 100%. This data can also be seen in Table 1 where N/A is used for the missing data and the totals columns when splitting was used because each stream is only capable of indicating the number of missed tweets and not which tweet IDS were missed. Since multiple crawlers may have overlap in the tweets that they do not receive, it is not possible to determine the number of unique tweet IDs missed across each crawler. Additionally, if limit track is an accurate metric, then the number of missed tweets for a single stream with

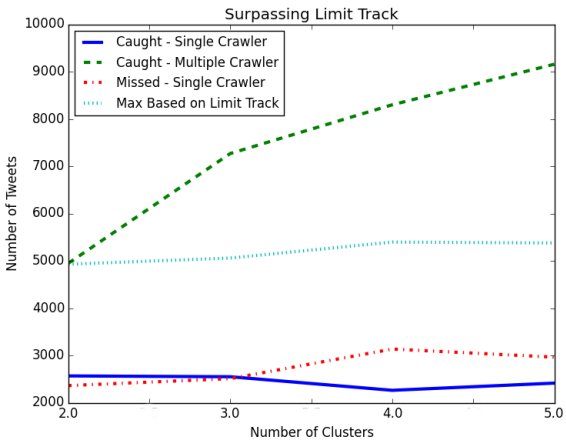

Figure 1: Using a single crawler it is possible to gather tweets from at most 400 keywords. As can be seen, the rate of tweets caught remains stable for a single crawler. Splitting the same keywords across multiple crawlers results in immediate improvement in the number of unique tweets caught as well as allowing the sampled population to go beyond the population size indicated by Twitter.

Table 1: Impact of Additional Crawlers on Sample Coverage. Since multiple crawlers may have overlap in the tweets that they do not receive, it is not possible to determine the number of unique tweet IDs missed across each crawler - we use N/A when this is the case.

	Caught	Missed	Total	Coverage
Unsplit	3632	4488	8120	44.7%
2-split	5060	N/A	N/A	62.3%
3-split	8714	N/A	N/A	107.3%
4-split	11143	N/A	N/A	137.2%

all possible keywords should indicate the total population. These figures provide strong evidence that the limit track reports supplied by Twitter are either inaccurate or they are an estimation.

In order to get the most tweets in a time period, we run multiple crawlers. Given a list of keywords, the Twitter streaming API will return every new tweet that contains any of the words specified. Therefore, by splitting the keyword list among multiple crawlers it becomes possible to gather tweets beyond the 1% limitation. Under perfect conditions, each additional stream increases the effective limit rate by 1%. Unfortunately, when partitioning keywords, it is important to keep in mind the possibility of overlap between the streams. For example, a tweet that contains keywords that were split between a number of crawlers will be duplicated in each stream. Tweet duplication in this manner reduces the overall efficiency of each stream. The stream splitting methods must be able to account for, and attempt to minimize, the potential for overlap between keywords.

In order to gather samples closer to the population size, we propose and evaluate three different splitting algorithms

- each with varying characteristics for the initial rate of growth and growth stability as additional crawlers are added:

- Round Robin
- Spectral Clustering
- K-Means Round Robin

4. EXPERIMENTS

Each of the following experiments is designed to test the efficiency of the given splitting method in obtaining a sample closer to the total sample population than is possible with the standard single stream method. The key factors that we will focus on include: speed of initial growth with a small number of crawlers, how stable the splitting method is for increasing growth as additional crawlers are added, and how many crawlers are required before we pass the population size estimation established by Twitter.

In each of the experiments, we drew from a pool of twenty-one streams. This allows us to use a single stream with all possible keywords as a baseline measure for standard gathering rate and population estimation with the limit track. The remaining twenty streams are then used for performing keyword splitting up to twenty ways. Each of these streams was able to track up to 400 keywords, the maximum number of keywords allowed by Twitter in any given stream. While twenty streams could easily track more than 400 keywords, we limit our search to all 400 keywords from a single stream split across up to twenty streams to allow for direct comparison with the performance of a single crawler. The keywords used were chosen by taking the most frequently occurring keywords from the Twitter sample stream in a ten minute period of time. These keywords contained a broad spectrum of topical words as well as multiple source languages. Keywords were chosen in this manner to ensure a high data rate from Twitter and represent a worst possible scenario for keyword splitting since a single keyword represents an atomic element that can not be further split. The set of keywords discovered in this fashion were used throughout.

For the purpose of these experiments, it was not necessary to track specific users or geographic coordinates simply because the volume of data obtained from these sources is minuscule in comparison to the top words used on Twitter. In cases where segmenting geoboxes is necessary, it is possible to segment it into any number of geographic regions while introducing no overlap between regions. In addition, the proposed solutions can be applied to the tracking of Twitter user names since they act as tokens in a similar manner to keywords. Limiting the number of keywords split between crawlers to the maximum capability of a single stream enables comparison by examining the change in sample population between each method while keeping the set of keywords constant. For example, using a single stream we know the number of tweets obtained, n, as well at the number of tweets left undelivered, m. Therefore, we should know the total sample size, N, such that $N = n + m$. Considering that the number of undelivered tweets is reported without any indication as to which data was lost, this total sample size is the only quantitative measurement given by Twitter as to the overall volume of the data. Furthermore, in cases where we want to employ multiple streaming accounts in a search strategy, the number of tweets missed, reported on a per-stream basis, becomes an unreliable measure due to the potential of separate streams to count a single missed tweet multiple times.

Each streaming agent captures text data from each tweet in order to create a word-to-word co-occurrence network. This data takes the form of word tuples followed by the number of times that the word pairs were observed across each tweet. The co-occurrence can then be used to create a network graph where each word is a graph node and the number of observations become undirected weighted edges. The resulting network graph G takes the standard form $G = (V, E, W)$ where each $v \in V$ is a word and each $e \in E$ is a co-occurrence observation with weight, W indicating the number of times a (v, v) pair was observed.

4.1 Experiment 1 - Round Robin

In order to reduce the initial overlap between crawlers, an additional "priming" step was added before each crawling experiment. Priming a search requires running an initial single level crawler for up to 10 minutes before creating any additional streams. During this stage the priming crawler observes all word pairs. Since words with a large number of pair observations are more likely to occur together, reducing the overlap between words will reduce the number of duplicated tweets. Though this priming step is a requirement for each splitting technique described here, it is possible to perform a very short initial priming search and subsequently update the clusters later as the word-to-word graph improves in quality. The priming step is not used in the results and is instead a method to obtain a word co-occurrence graph to be used in performing the initial splits.

The round robin method of stream keyword splitting is an effective baseline for other splitting methods as it is a straightforward method that requires very little additional processing power. Sampling the amount of tweets gathered and missed at each split level requires running one baseline stream that contains all selected keywords as well as k additional streams that contain the keywords split between each stream. While it is possible to sample all split levels simultaneously, the number of required accounts for a test of this type is $x_k = \frac{k(k+1)}{2}$ where k is the number of split levels and x is the number of accounts. Sampling all splits up to a split level of 7 would require 28 separate account which is unfeasible for our purposes. Additionally, the processing power required to maintain the set of unique tweet IDs for each stream becomes problematic very quickly. Alternatively, using a single baseline stream that contains all keywords and comparing the results to each split level independently requires a much lower number of accounts, $x_k = k+1$. It is this latter method that we use for each stream splitting analysis. At the completion of the priming stage, the word pairs, from the most frequently occurring to the least frequently occurring, are assigned to streams in a round robin fashion. Each split level runs for 30 minutes before resetting all streams, including the baseline stream, and beginning the next split level. Resetting the baseline stream is key to analyzing each stream level in this method as it allows a comparison of the difference between a single stream and multiple split streams over a given window of time and thereby making it unnecessary to maintain all data for every level of split at once.

The graph shown in Figure 2 show that we were able to eclipse the limit track maximum by 12 splits at which point we were able to gather six times as many unique tweets containing proper keywords than was possible with a single stream. Reaching 20 split levels nearly doubled the number of unique tweets gathered over the maximum indicated by

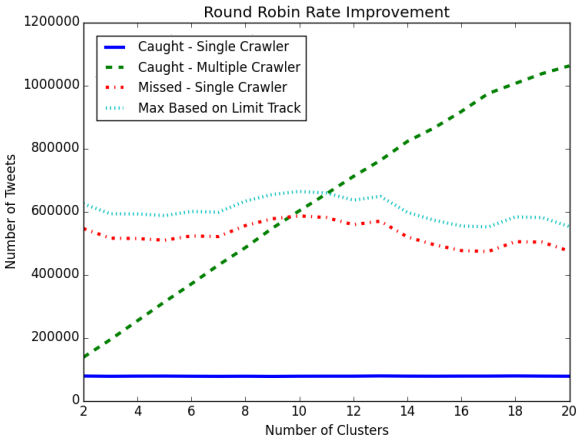

Figure 2: Round robin splitting based on word co-occurrence tends to show a steady rate of gain as additional crawlers are added.

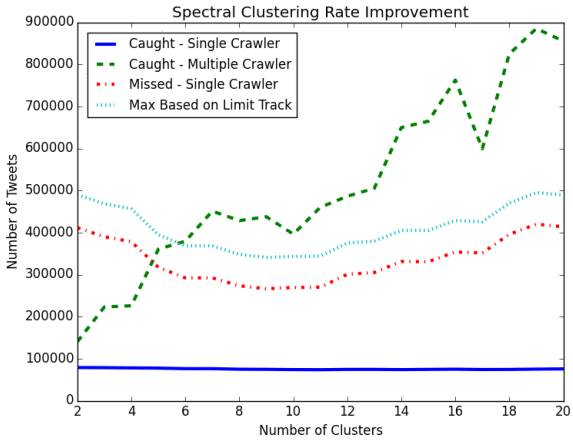

Figure 3: The results of Spectral Clustering do show an increase of sample coverage overall. However, the clustering creates unbalanced splits where one stream, while still a good cluster, may contain significantly more words than others. The lack of balance manifests through instability in the rate of gain from each additional crawler.

Twitter. Constructing round robin splits can be found in Algorithm 1.

Algorithm 1: Round Robin Splits Construction

```
input  : graph G, num_clusters k
output: array of lists
for node v in G(V, E) do
  | keywordList += v
end
sortedList := Sort(keywordList by occurrence rate);
for word, index in sortedList do
  | listNum := index % k;
  | assign word to split list k
end
```

4.2 Experiment 2 - Spectral Clustering

Spectral clustering, an extension of K-Means clustering which performs "low-dimension embedding" [20], directly leverages the word occurrence graph. Unlike K-Means, spectral clustering requires an affinity matrix in order to create accurate clusters based on how the items relate to one another. This clustering method allows us to define a number of clusters, k, and the spectral clustering algorithm will incorporate the nuances of the similarity between the items in order to improve cluster results. Like most clustering algorithms, spectral clustering does not make any guarantee on the size of each cluster. As a result, cluster size can vary to a large degree which has implications for the usefulness of this method.

The combined effect of these properties of spectral clustering manifest themselves as a number of interesting properties. First, as the number of requested clusters increased, keywords in each sub cluster also became increasingly biased towards individual languages. This is a favorable trait for reducing overlap since two or more streams gathering keywords of differing language should have a low rate of word overlap except in the case of multi lingually authored tweets. Secondly, despite the ability to specify the number of clusters, if the underlying similarity does not lend itself well to a high number of partitions a few of the resulting clusters will be very large while many will be small. Though this behavior is preferable for most applications that employ clustering, a significant difference in the size of clusters causes some streams to significantly over perform the 1% limit and subsequently become severely rate limited while streams based on smaller clusters fail to reach a limiting rate at all. This effect can be seen in Figure 3. Clustering based on word occurrence quickly passes the Twitter limit with only 6 streams active but shortly thereafter struggles to gain much ground. Wild fluctuation can be observed between each split level and while there is overall growth it is possible to gather a smaller sample with a larger split level. Such inconsistencies were not observed in Figure 2 further indicating how detrimental sensitivity to cluster size is when considering methods for gathering tweet samples. Spectral clustering based splits can be found in Algorithm 2.

4.3 Experiment 3 - K-Means Round Robin

The K-Means Round Robin (KMRR) approach looks to incorporate the strengths of both previous splitting methods. It borrows the near equivalently sized groups found in the round robin method which enable stable growth across split levels and the use of intelligent clustering from the spectral clustering method to reduce tweet duplication, or overlap. Rather than using spectral clustering, however, we use K-Means clustering [15]. In order to accomplish this type of non-standard clustering with K-Means we first need to convert the network graph into a dissimilarity matrix. This can be done by constructing the standard network graphs weighted adjacency matrix and normalizing by the highest occurring word pair. Next, we use the dissimilarity matrix in a process called Multidimensional Scaling, or MDS [3]. Multidimensional Scaling uses the computed dissimilarity values to transform data into another graph space - in this case two dimensions. MDS leverages the dissimilarity between each point as a measurement distance and seeks to find an embedding in the new dimensional space that maintains these

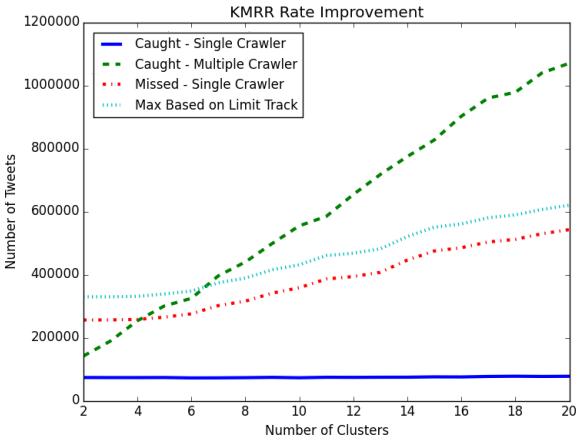

Figure 4: KMRR splitting displays quick initial gains with a low number of crawlers while eventually settling into a steady growth rate as additional crawlers are added.

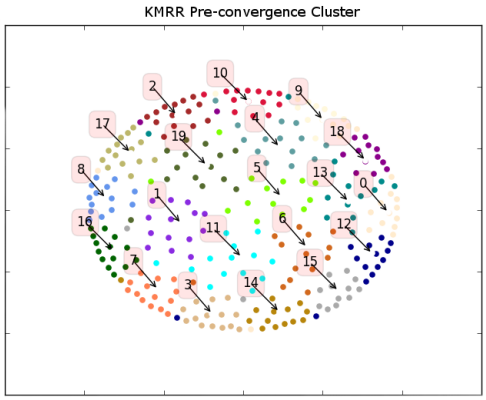

Figure 5: After performing Multidimensional Scaling on the data set, the words are clustered according to K-means. The discovered cluster centroids are then used to create balanced groupings by assigning a centroid to the closest unassigned word in a round robin manner. However, this process introduces intruding words as the number of remaining assignments decreases.

Algorithm 2: Spectral Clustering Splits Construction

```
input  : graph G, num_clusters k
output: array of lists
matrix A := [][];
for node v in G(V, E, W) do
 |  wordIDs[v] := unique ID
end
/* create affinity matrix for spectral
   clustering                                   */
for word pair (v1,v2) in G(V, E, W) do
 |  Construct symmetric matrix A for all words such
 |  that v1,v2 := E.weight;
end
labels := the result of Spectral Clustering with k
clusters and the matrix A;
for label, cluster in labels do
 |  splitLists[cluster].append(label)
end
```

distances as closely as possible. With our network newly transformed into a two dimensional graph, K-Means is run on the data to find centroid locations. It is important to note that we do not use the learned K-Means labels as this would not satisfy the requirement for relatively balanced keyword splits. Instead, keywords are assigned to clusters in a round robin fashion based on euclidean distance from the centroid. The combination of MDS transformation with round robin k-means centroid assignments is shown in Figure 5.

It is obvious to see that assigning words to centroids in a round robin style introduces intruding words into clusters. Accounting for word intrusion is accomplished through a convergence step. The convergence process examines the items assigned to each centroid and swaps the centroid assignment for any pair of items where doing so would reduce the average distance to each centroid following the inequality, $\frac{b_{c1}+a_{c2}}{2} < \frac{b_{c2}+a_{c1}}{2}$, where $c1$ and $c2$ are cluster centroids and the remaining variables follow the notation that a_{c1} is the distance to centroid 1 from item a. This process is repeated until no improvement can be found between any pair of nodes. The convergence process is depicted in Figure 6. While performing the convergence step would normally be time consuming, the number of items clustered is never larger than 400 items. Figure 7 shows the final cluster assignments after the completion of the convergence step. These clustering assignments are completely absent of intrusion. Using this method the rate at which multiple crawlers exceed the limit track becomes comparable to spectral clustering at 7 splits as can be seen in Figure 4. Additionally, K-Means Round Robin does not suffer from the problem of inconsistency between split levels showing consistency similar to that found in the round robin experiment. KMRR-based splits can be found in Algorithm 3.

A comparison of each splitting method can be seen in Figure 9. These results, shown also in 2, are the average coverage rates related to the total sample size estimated from the limit track over 20 trials. Though the Twitter limit track is not a perfect indicator of the total population of data available from a complete set, it is used in this comparison as a relative measure as opposed to an absolute. Given additional crawlers, it is very likely that the limit track would again be eclipsed. Each splitting method was able to produce a sample significantly closer to the total population as estimated by Twitter and sometimes many times larger than a single stream.

5. CLUSTER REALIGNMENT

The next step is determining if there is a difference between results when the clusters are computed at some time variable above the reconnection limit imposed by Twitter. To accomplish this, we use one stream that runs indefinitely and constantly updates the observed weight for each word occurrence pair. This stream does not maintain a tweet set in order to keep the memory footprint from growing too large. Every 1000 seconds the cumulative network is clustered and new splits are designated for every stream.

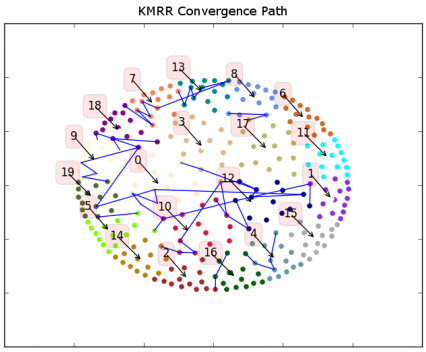

Figure 6: The convergence step maintains group size by operating on pairs of nodes with differing centroid assignments. On each pass, if a pair of nodes is discovered for whom swapping them will reduce the average distance between centroids then it is considered a good swap and immediately performed. The lines between each node indicate the swap path as a pair of cluster assignments gradually improve.

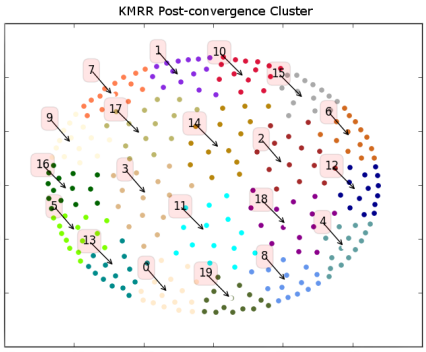

Figure 7: At the completion of the KMRR convergence step, the resulting clusters can be seen to have no intruding assignments while maintaining the balanced group size between clusters.

This time step was chosen in order to avoid connection rate limiting from Twitter which will block connections from a stream if it attempts to reconnect more frequently than every 15 minutes. Unfortunately, Twitter does not supply any method for changing the search parameters of a stream without performing a reconnection so any method of cluster realignment will be unable to perform these steps more frequently than the 15 minute interval.

Cluster realignment seeks to improve the performance of the spectral clustering method as well as KMRR. Since one stream is dedicated solely to maintaining and updating the word-occurrence network, performing either of the cluster-based splitting methods at a later period of time should improve the resulting clusters. Using cluster realignment, it becomes possible to run a very short priming step or even eliminate the priming step altogether. This allows the streams to begin gathering data immediately while also reducing the overlap between each stream over time. The effect of cluster realignment can be seen in Figure 8. While the initial data

Algorithm 3: *K*-Means Round Robin Splits Construction

```
input  : graph G, num_clusters k
output: array of lists
matrix dissimilarity := [][];
counter := 0;
for node v in G(V, E, W) do
  | wordIDs[v] := unique ID
end
max := compute highest rate of occurrence for
normalization;
/* create dissimilarity C matrix for MDS      */
for word pair (v1,v2) in G(V, E, W) do
  | Construct symmetric dissimilarity matrix for all
  | words such that v1,v2 := | E.weight − max | ;
end
wordGraph is the result of Multidimensional Scaling of
the dissimilarity matrix to 2 dimensions;
labels := KMeans(num_clusters = k, data =
wordGraph)
for i := 0 to number_of_keywords do
  | assign keyword i to the closest cluster centroid
end
/* enter convergence step                     */
while any improvement can be found do
  | for i := 0 to number_of_clusters do
  |   | for j := i to number_of_clusters do
  |   |   | if swapping a point in each cluster would
  |   |   | reduce the average distance to each centroid
  |   |   | then
  |   |   |   | swap cluster assignments
  |   |   | end
  |   | end
  | end
end
for label, cluster in labels do
  | splitLists[cluster].append(label)
end
```

gathering rate using this method can be somewhat unstable, over a period of time the number of tweets gathered per second does become stable and shows an overall increase. Over larger periods of time, the word occurrence network becomes increasingly stable, and, as a result, the overall increase of performing realignment becomes less efficient. This is especially true when considering that performing clustering, disconnection, and reconnection can take some time and introduce small gaps into the dataset. Twitter streams also do not return immediately to their highest potential data rate which introduces a dip in the number of tweets obtained for a small period of time. Overall, cluster realignment improves the possible data rate but these factors should be considered when implementing a realignment scheme.

6. LANGUAGE CLUSTERING

In spectral clustering based on word co-occurrence, about 72% of the largest clusters were one language with a mix of words from other languages. When k = 20, many of the smaller clusters were completely biased towards their language to the point that they were always very close to, if not completely from, a single language. The high rate of

Figure 8: Performing cluster realignment by maintaining and updating the word co-occurrence graph over a long period of time can improve the number of tweets gathered per second without the need to introduce additional crawlers. Initially, the lack of a significant sample for the co-occurrence graph causes fluctuations in the gathering rate but as the sample size increases the volume of data per second becomes stable and is still able to maintain a constant rate of growth even as the amount of data in the population (limit maximum) reduces.

strong language clusters is a good indicator of correct cluster assignments. In all methods, a high rate of language clustering should result in searches with less overlap. Future methods for search improvement should seek a balance between language clusters, word co-occurrence overlap reduction, and maximum per-crawler stream utilization.

KMRR also displayed this property by produced heavily language-biased clusters but, as is likely the result of the round robin process, few of the clusters were completely from a single language. There is a potential for optimization within the KMRR clustering process where by producing slightly less balanced clusters in exchange for improved language clustering may produce better results. On the far end of the spectrum, the round robin method seemed to produce a completely mixed set of language keywords as is expected. Further experiments would need to be run with attention paid particularly to the language bias from these clustering methods in order to obtain a better understanding of the effect of language clusters on streamed search results.

7. CONCLUSION

We ask whether the volume of data missed, as reported by the Twitter limit track system, is an accurate and useful measurement for determining sample size. Furthermore, we ask whether standard methods used for gathering streamed data from the Twittersphere can be improved to increase coverage for a gathered dataset. To answer these questions, we used a battery of twenty streams as well as a single comprehensive baseline stream to gather tweets for a large number of high volume keywords. We provide a methodology for comparing relative improvement in sample size in the absence of reliable reporting as well as a series of algorithms capable of a multiplicative increase in sample coverage.

We started our analysis by inspecting the performance of the Twitter limit track process in single and multi-stream situations. It has been assumed that the Twitter limit track provides an accurate metric for the overall size of a dataset with a given set of keywords. When using a single stream, the limit values returned, in addition to the number of tweets gathered, should be the size of the complete data set if it were possible to obtain all 100%. However, the simple addition of extra streams operating on subsections of the same keywords were able to quickly overtake the hypothetical maximum volume of unique tweets. Such ease in producing beyond the maximum limit as calculated by Twitter cast serious doubts on the validity of the limit track system. In the absence of a well-defined baseline we employed a relative method to measure improvement obtained through alternative streaming methods.

Next, we proposed a series of methods for separating keywords in order to obtain the best possible sample coverage. The simplest proposed method, Round Robin splitting, displayed stable sample growth with the addition of each stream and was able to overtake the Twitter-calculated complete set. The stability of growth indicated the importance of balancing keywords across streams - a property that would later be employed in K-Means Round Robin.

The second method, Spectral Cluster splitting, again employed a word co-occurrence graph to build a similarity matrix. The clusters obtained in this manner tended to vary significantly in size but showed interesting properties, such as a tendency for languages to cluster together. Spectral Cluster-based streams showed the sharpest rate of initial growth with a smaller number of streams. However, when the number of streams grew large the volume of increase with each stream became unstable as a result of small clusters failing to fully populate the stream bandwidth.

Using the lessons learned from Round Robin splitting and Spectral Cluster splitting, we proposed a balanced solution in K-Means Round Robin. KMRR uses K-Means clustering before following a convergence process to produce balanced cluster sizes while maintaining as much of the original clustering properties as possible. KMRR showed the rapid initial growth displayed by Spectral Clustering as well as the stability obtained from Round Robin splitting. While clusters still displayed a tendency to group by language, each cluster had a higher rate of out-of-language word intrusion than was seen in the Spectral Clustering results. Each of these methods allow for cluster re-computation on the fly, and improvements to the rate of tweets obtained per second were seen when periodically realigning from an improved word co-occurrence graph.

There are many interesting possible extensions to this work. Since it can be shown that the Twitter limit track is not a true indication of the overall population for a given search, further inquiry into the methods used for calculating the limit may reveal interesting features of the Twitter tweet dissemination process and provide insight to the source of bias observed in Twitter streams. While our experiments focused on a single feature of each keyword, the co-occurrence rate, the language clustering side effect indicates that there are potentially other features that may introduce sources of overlap. An example of this would be the semantic meaning between each keyword where there may be potential for further reduction of overlap by seperating keywords with similar meaning. Identification of other such features could

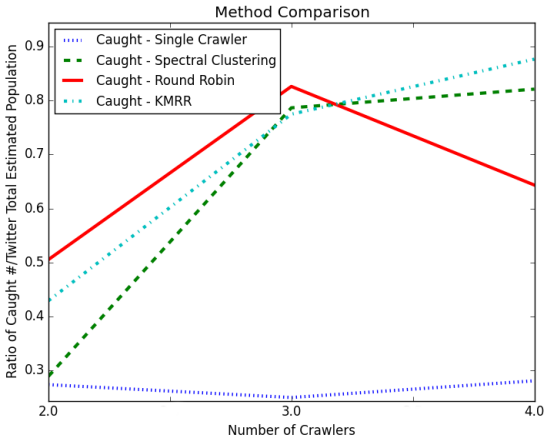

Figure 9: This graph shows the average sample coverage for a given set of search keywords across 20 trials. Each of our sampling methods significantly outperform the single stream with some variation in characteristics between each method.

Table 2: Sample Coverage of Total Population

	Unsplit	2-split	3-split	4-split
Round Robin	19.02%	50.54%	82.58%	64.34%
Spectral Clustering	19.02%	28.95%	78.63%	82.08%
KMRR	19.02%	42.93%	77.45%	87.63%

further strengthen the gathering process and lead to better and less biased samples.

8. ACKNOWLEDGEMENTS

This work is sponsored, in part, by Office of Naval Research grant N000141410095.

9. REFERENCES

[1] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu. Predicting Flu Trends using Twitter Data. In *INFOCOM*, pages 702–707. IEEE, 2011.

[2] S. Asur and B. A. Huberman. Predicting the Future with Social Media. In *WI-IAT*, volume 1, pages 492–499. IEEE, 2010.

[3] I. Borg and P. J. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media, 2005.

[4] A. Ceron, L. Curini, S. M. Iacus, and G. Porro. Every Tweet Counts? How Sentiment Analysis of Social Media can Improve our Knowledge of Citizens' Political Preferences with an Application to Italy and France. *New Media & Society*, 16(2):340–358, 2014.

[5] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. CRC press, 1994.

[6] M. Garcia-Herranz, E. Moro, M. Cebrian, N. A. Christakis, and J. H. Fowler. Using friends as sensors to detect global-scale contagious outbreaks. *PLoS ONE*, 9(4):e92413, 04 2014.

[7] S. Ghosh, M. B. Zafar, P. Bhattacharya, N. Sharma, N. Ganguly, and K. Gummadi. On Sampling the Wisdom of Crowds: Random vs. Expert Sampling of the Twitter Stream. In *CIKM*, pages 1739–1744. ACM, 2013.

[8] J. Gomide, A. Veloso, W. Meira Jr, V. Almeida, F. Benevenuto, F. Ferraz, and M. Teixeira. Dengue Surveillance Based on a Computational Model of Spatio-temporal Locality of Twitter. In *WebSci*, pages 1–8. ACM, 2011.

[9] S. González-Bailón, N. Wang, A. Rivero, J. Borge-Holthoefer, and Y. Moreno. Assessing the Bias in Samples of Large Online Networks. *Social Networks*, 38:16–27, 2014.

[10] 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*, pages 75–83. Springer, 2014.

[11] D. Kerg, R. Roedler, and S. Seeber. On the Endogenesis of Twitter's Spritzer and Gardenhose Sample Streams. In *ASONAM*, pages 357–364, 2014.

[12] S. Kumar, G. Barbier, M. A. Abbasi, and H. Liu. TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief. In *ICWSM*, pages 661–662, 2011.

[13] R. Li, S. Wang, and K. C.-C. Chang. Towards Social Data Platform: Automatic Topic-focused Monitor for Twitter Stream. *VLDB*, 6(14):1966–1977, 2013.

[14] Y. Lu, F. Wang, and R. Maciejewski. Business Intelligence from Social Media: A Study from the VAST Box Office Challenge. *Computer Graphics and Applications, IEEE*, 34(5):58–69, Sept 2014.

[15] J. MacQueen et al. Some Methods for Classification and Analysis of Multivariate Observations. In *BSMSP*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[16] L. Madlberger and A. Almansour. Predictions Based on Twitter-A Critical View on the Research Process. In *ICODSE*, pages 1–6. IEEE, 2014.

[17] F. Morstatter, N. Lubold, H. Pon-Barry, J. Pfeffer, and H. Liu. Finding Eyewitness Tweets During Crises. *ACL*, pages 23–27, 2014.

[18] F. Morstatter, J. Pfeffer, and H. Liu. When is it Biased?: Assessing the Representativeness of Twitter's Streaming API. In *WWW*, pages 555–556, 2014.

[19] 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. In *ICWSM*, pages 400–408, 2013.

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *NIPS*, pages 849–856. MIT Press, 2001.

[21] Z. Tufekci. Big Questions for Social Media Big Data: Representativeness, Validity and Other Methodological Pitfalls. In *ICWSM*, pages 505–514, 2014.

[22] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. *ICWSM*, 10:178–185, 2010.

[23] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness. In *CHI*, pages 1079–1088. ACM, 2010.