Mining Twitter Data with Resource Constraints

George Valkanas, Ioannis Katakis, Dimitrios Gunopulos
University of Athens
{gvalk,katak,dg}@di.uoa.gr
Antony Stefanidis
George Mason University
astefani@gmu.edu

***Abstract*—Social media analysis constitutes a scientific field that is rapidly gaining ground due to its numerous research challenges and practical applications, as well as the unprecedented availability of data in real time. Several of these applications have significant social and economical impact, such as journalism, crisis management, advertising, etc. However, two issues regarding these applications have to be confronted. The first one is the financial cost. Despite the abundance of information, it typically comes at a premium price, and only a fraction is provided free of charge. For example, Twitter, a predominant social media online service, grants researchers and practitioners free access to only a small proportion (1%) of its publicly available stream. The second issue is the computational cost. Even when the full stream is available, off the shelf approaches are unable to operate in such settings due to the real-time computational demands. Consequently, real world applications as well as research efforts that exploit such information are limited to utilizing only a subset of the available data. In this paper, we are interested in evaluating the extent to which analytical processes are affected by the aforementioned limitation. In particular, we apply a plethora of analysis processes on two subsets of Twitter public data, obtained through the service's sampling API's. The first one is the default 1% sample, whereas the second is the Gardenhose sample that our research group has access to, returning 10% of all public data. We extensively evaluate their relative performance in numerous scenarios.**

I. Introduction

Web mining is an ever-changing discipline, as the web itself evolves over time. Social media are an integral part of today's web ecosystem, and provide numerous opportunities for applications with significant social and economical impact. Characteristic examples include computational journalism and psychology, crisis management, resource allocation, advertising, etc. At the same time, interesting research questions lie at the heart of such applications, which need to be addressed efficiently and effectively. For instance, computational journalism requires high quality data, in order to provide credible information. Crisis management and event detection call for real-time information processing, so that one can assess and respond to the situation quickly and judiciously.

In addition to the algorithmic challenges, applications that rely on social media are typically confronted with two issues:

- **Financial cost**: Despite the abundance of information, and regardless of the way in which the data is used, it typically comes at a premium price, with only a fraction being free of charge. For instance, Twitter grants researchers and practicioners free access to only a small sample (1%) of its publicly available stream. Licences for other sample sizes, such as the Gardenhose and the Firehose, which provide 10% and 100% respectively, are costly and difficult to obtain.
- **Computational cost**: Even when the full stream is available, off the shelf approaches are unable to operate in such settings due to the real-time computational demands. For example, Twitter generates 7 Giga Bytes of data per minute. This data rate makes the analysis process quite cumbersome. Consequently, real world applications as well as research efforts that exploit such information are, in practice, limited to utilizing only a subset of the available data.

Taking into account the issues highlighted above, those who engage in social media analytical tasks have practically no choice but to resort to the downsized information. However, being only a small fraction of the entire stream, it is unclear how reliable this information is for each type of application. A recent first effort towards this research direction is described in [17]. In that work, Morstatter et al. compare the default 1% sample against the 100% Firehose sample, and the comparison spans through various tasks. Given that they had access to the entire stream, one of the main results of their work is that the received 1% sample is not a uniform random sample.

Similar in spirit to the work of [17], we, too, are interested in evaluating the extent to which analytical processes are affected by the aforementioned limitation, i.e., having access to a limited proportion of the entire information. In particular, we apply a plethora of analysis processes on two subsets of Twitter public data, obtained through the service's sampling API's. The first one is the default 1% sample, whereas the second is the Gardenhose sample that our research group has access to, returning 10% of all public data. We extensively evaluate their relative performance in numerous scenarios. The difference between our current work and the one in [17] is that we focus on specific aspects of the data, namely spatial and temporal, which are inherent due to the nature of the medium itself, as we explain in the following.

Given that it would be impossible to apply all data analytical techniques in order to evaluate extensively the obtained samples, we select a representative subset, motivated by different application scenarios, and report on those findings. More specifically, we answer the following research questions:

- *Sentiment Analysis*: Sentiment analysis has been used to evaluate the performance, and predict the outcome of potical debates and elections [6], [16], to perform brand monitoring and event detection [24], [15], to name a few. Consequently, we want to study how the Twitter API sampling affects the widely used spatio-temporal analysis task of Sentiment Analysis.
- *Geo-located information* - How many geo-located tweets are provided by the Streaming API and the Garden Hose? How is the relation between these two values varying through space and time? In all of these tasks we study how the difference between the two streams varies in different locations. Stefanidis et al. [21] reported that approximately 16% of the Twitter feeds in their experiments had detailed location information with it in the form of coordinates, while another 45% of the tweets they collected had some geo-location information at coarser granularity (e.g. the city level).
- *Popular tweets* - We extract trending topics of various locations using the Twitter API and the Garden Hose and study their differences.
- *Social Graph Evolution* - We focus on the retweet graph, and want to see how the sampling process affects certain of its measures.
- *Linguistic Analysis* - We apply language detection, to identify whether the received samples exhibit statistical properties which are known to hold in the real world. More specifically, we answer the question whether the written languages found in Twitter are a representative sample from languages in the physical world.

II. Related Work

Twitter is one of the predominant social media sites in today's webosphere. Its real time nature and short-form communication distinguish it from the other networking services. These inherent characteristics have made it a primary source of information in real-time event detection [5], [26], [24], monitoring [15], [4] and crisis management situations [20], [7], [22]. The medium grew in popularity and recognition as it played a pivotal role in news and information broadcasting during the Middle-eastern crisis, and has been also used extensively to evaluate the performance of political candidates and their campaigns [16], [6].

A. Spatio-temporal analysis of Twitter feeds

The service is also characterized by the diversity of its users, in terms of location, spoken languages, backrgound, interests, etc. The spatial aspect is of paramount importance for a large number of applications, such as event detection and response, targeted advertising and community detection to name a few. Towards that end, users are able to geo-tag their tweets, i.e., attach GPS information.

Unfortunately, despite providing high precision information, it has been shown in numerous studies [21] that the percentage of GPS-tagged tweets is too low. Moreover, such information is typically mediated through other location-based services, e.g. Foursquare[1], which reduces the textual content provided by the users themselves. To address this shortcoming, researchers have proposed techniques which aim to extract spatial information from the text, either of the tweets or the users' profiles [1], [2], [11], [23].

B. Sampling Social Data Streams

The number of users who are actively using the service, and the amount of information posted daily are enormous. To cope with these sizes, sampling is usually employed to downsize the data, while maintaining the properties of interest. For example, the work in [10] applied online sampling of the social graph. This approach is equivalent to a uniform sampling of the nodes, without knowing the entire graph in advance. The authors in [9] apply sampling on users, in order to identify topical experts.

The work most closely related to ours is [17]. Having access to the Firehose, the authors compare the default sampling policy of Twitter against the entire Twitter stream. One of the main outcomes of their work was that the sample provided through Twitter's default streaming API is not a random sample. Compared with that work, we want to evaluate the performance of the 10% sample (Gardenhose) and contrast it with the 1% default sample. We also take a more temporal standpoint of evaluation, and focus on more analytical processes, such as sentiment analysis and linguistic analysis. We are also interested in evaluating properties of retweeted posts, that go beyond the retweet graph itself. Nevertheless, the reported values in [17] can be used as ground truth information, considering that they had access to the full Twitter stream.

III. The Data

Our experimental setup relies on data received from the Twitter service. Contrary to techniques typically used for harvesting data from online resources, Twitter provides a streaming API[2], which resembles the publish-subscribe paradigm: users subscribe to the service, with a request to receive data. Twitter sends the data to the subscribed users, according to a sampling policy. This results in the service being less stressed by contiuous probes for new data.

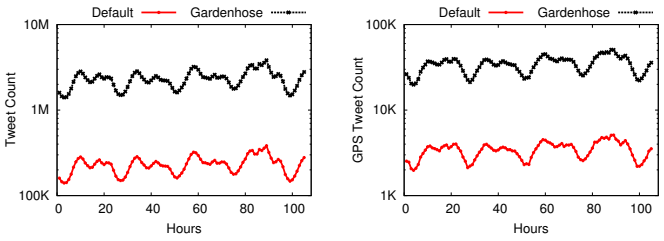
The default sampling policy returns 1% of all publicly available tweets, i.e., tweets from users who have allowed everyone to see their posts. Our group has also been granted access to the Gardenhose, which returns a 10% sample of the publicly available tweets. In both cases, the sampling policy is controlled entirely by Twitter.

Our main set of experiments is conducted on two datasets, obtained by crawling the 1% and 10% of the service over the same period of time. We monitor the Twitter stream for slightly over a 4-day period in November 2013, and store the tweets as they are provided. We subsequently perform our analyses in an offline fashion, using a custom workflow infrastructure [25].

Figure 1a shows the amount of information we collected within each hour from the onset of our experiment, for both sampling policies. An immediate observation is that the two

[1] https://foursquare.com/

[2] https://dev.twitter.com/docs/api/1.1/get/statuses/sample



(a) All tweets (b) GPS-tagged tweets

Fig. 1. Comparing default and gardenhose samples for volume over time

samples differ by an order of magnitude, which is expected given the sampling percentages offered by the service. Secondly, we see that both samples exhibit the same temporal pattern, with the same increases / decreases appearing in both streams. Finally, it is interesting to note that in both cases, there is a certain periodicity in the data, which coincides with the 24-hour cycle of a day.

IV. Geo-location Coverage

An important aspect of Twitter data is that several of them are geotagged, meaning that the posting user has attached a GPS-quality signal to the tweet when uploading the information. Such information can be particularly important to understand where the user is and what they are refering to.

Figure 1b shows the number of geotagged tweets that were received from the two Twitter sampled streams, the default one (red) and Gardenhose (black). It is interesting that we observe the same temporally periodic pattern as the one in Figure 1a. Moreover, the geotagged tweets are between 1-2% of their respective raw sampled data, and the two streams (of geotagged tweets) differ by an order of magnitude, which is a result of the Gardenhose returning 10× more tweets that the default sample. Finally, several of the fluctuations that we observed in Figure 1a have been flatened out when we consider the geotagged tweets alone.

Twitter also allows its users to ask for geotagged information. In this case, a different approach is used to connect to the publish-subscribe mechanism, indicating that geotagged tweets are requested. The user provides a bounding box, by specifying 4 coordinates in the form $[(lat_{min}, lon_{min})(lat_{max}, lon_{max})]$, and Twitter returns tweets that fall within this region. In this particular case, where geotagged tweets are asked for instead of a general sample, the volume of the returned results is the same for the two samples. The reason is due to the different mechanism used in this case by the service. We omit the respective figures due to space restrictions.

However, we were interested in other types of differences that may arise by using this mechanism. To this end, we focused on a particular region in London, and applied different bounding boxes, which slightly overlap. Table I shows the coordinates for the bounding boxes that were used, along with the number of tweets that were received, whereas Figure 2 visualizes these on a map.

Table II shows the similarity between the collected tweets. In particular, we have computed the Jaccard similarity of the

TABLE I. Description of the the GPS-driven crawls

ID	Bounding Box	#Tweets
Crawl1	[(-0.1754, 51.4830), (-0.0704, 51.5327)]	35275
Crawl2	[(-0.2654, 51.4830), (-0.1604, 51.5327)]	27811
Crawl3	[(-0.2254, 51.4830), (-0.1204, 51.5327)]	27811
Crawl4	[(-0.1854, 51.4830), (-0.0804, 51.5327)]	27811

Fig. 2. Bounding boxes of Table I, Crawl1: Green, Crawl2: Orange, Crawl3: Red, Crawl4: Blue

received data and report these values. It is interesting that the measured similarity is generaly quite high, even when the overlap is low (e.g., Crawl 1 and 2). As the overlap increases between the bounding boxes that we applied, so does the similarity between two different crawls.

TABLE II. Jaccard Similarity between the GPS-driven crawls

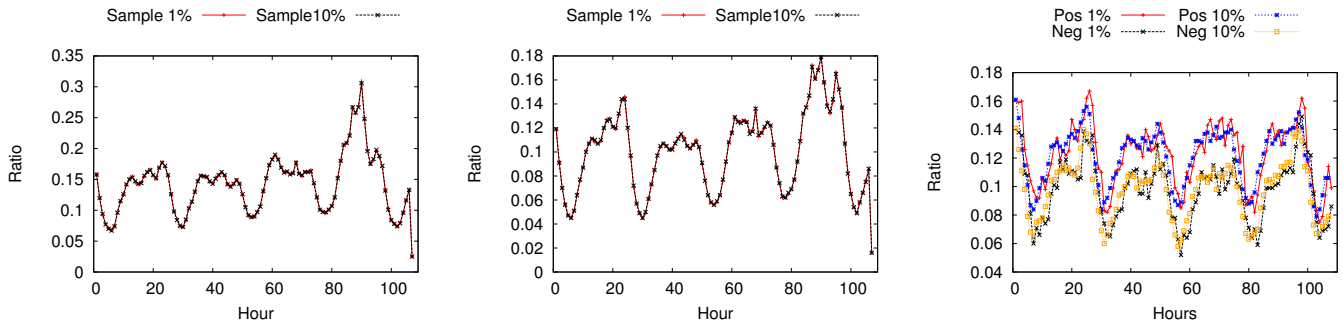
	Crawl1	Crawl2	Crawl3	Crawl4
Crawl1	1.0	X	X	X
Crawl2	0.527	1.0	X	X
Crawl3	0.671	0.788	1.0	X
Crawl4	0.866	0.612	0.777	1.0

Figure 3 shows how the 4 distinct bounded-driven crawls performed over time for a single day. With some minor fluctuations, we observe that all of them follow the exact same pattern. Note that the first half-hours, where there is a steep decline in volume, are in the early hours of the day because the crawl was started around 10:30pm. Therefore, the x-values between 3 and 15 depict the volume between midnight and 8 o'clock in the morning.

V. Sentiment Analysis

Sentiment analysis is probably one of the most frequent tasks applied on Twitter [13], [16], [14], [18]. The vast availability of opinions expressed in Twitter raised the interest of the research community as well as the industry. Hence we consider this problem as one of most critical tasks where the sufficiency of the Streaming API (1%) should be evaluated.

In general, the problem of sentiment analysis is that given a text segment t_i, it is requested to assign it into one of the polarity classes (`negative, neutral, positive`) according to the sentiment that it expresses. In fact, most of



(a) Positive Sentiment (b) Negative sentiment (c) Sentiment in geotagged tweets

Fig. 4. Comparing tweets with sentiment

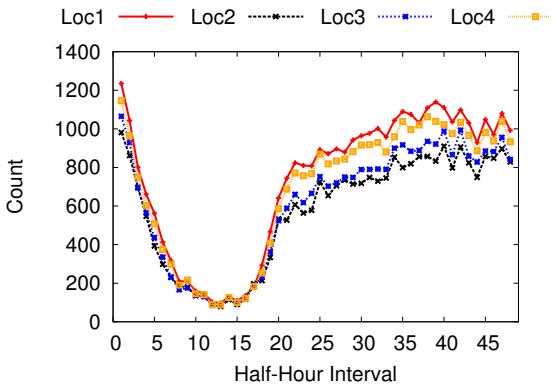


Fig. 3. Different crawls from London

the times, the output is a *sentiment rate* in $[0, 1]$. In the setting that we consider, the task is to assign such a label to each tweet individually.

For our analysis, we employed a lexicon-based approach, whereby positive opinion words contribute towards the positive classification of the text whereas negative opinion words contribute towards negative classification. The obvious advance of a lexicon-based approach is that no training data are required and that there are low computational requirements. Naturally, to apply such a technique, two sets of words are required: a positive and a negative one. We utilize the lists provided in [12]. Given the text of a *tweet*, we count how many words appearing in it express positive (w_+) or negative (w_-) opinion. We then assign the tweet to one of the classes as follows:

$$sentiment(tweet) = \begin{cases} positive & : |w_+| > |w_-| \\ negative & : |w_+| < |w_-| \\ neutral & : otherwise \end{cases}$$

Figure 4 reports the ratio of positive and negative tweets, per hour, over all tweets received during the same time period. Interestingly, we observe that the ratio of tweets is the *same* in both occasions, although the absolute values differ by an order of magnitude. The ratio is higher for positive tweets, with certain cases having twice as many positive tweets. There is also periodicity in the data, similar to the one that we observed in previous sections.

Inherently, Sentiment Analysis has spatio-temporal characteristics. In the last USA presidential elections, many organizations kept track of the sentiment *during* the electoral period (trend) for *each one of the states*. For this reason, we also provide an experimental comparison by applying sentiment analysis to the subset of geotagged tweets that were received with the two sampling policies.

As we observe from Figure 4c, the ratios of positive and negative geotagged tweets exhibit similar patterns to the general stream, shown before. Even in geotagged tweets, there are more positive ones than negative, regardless of the streaming policy used. The ratios, however, are in principle lower than in the general stream, meaning that geotagged tweets offer less sentiment-oriented information.

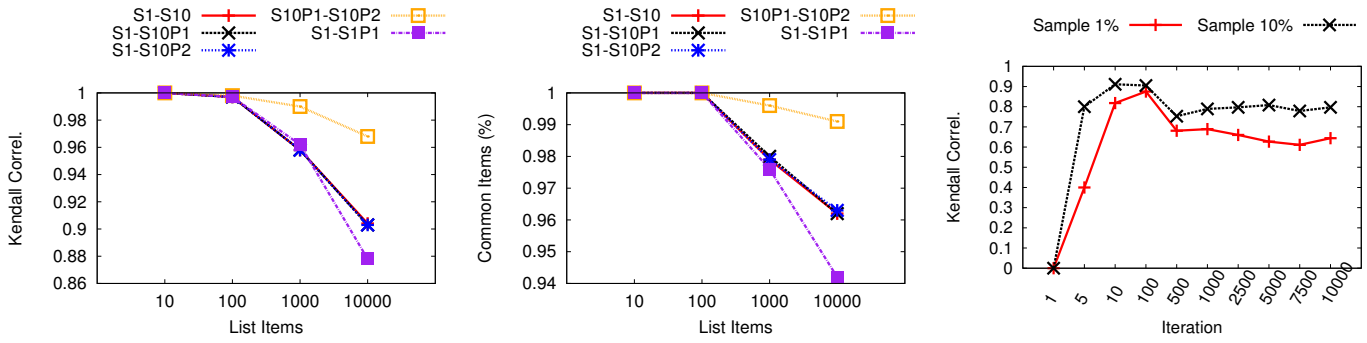
VI. Popular Topic Detection

One of Twitter's most characteristic features is the ability of its users to *retweet* other posts. Such an action allows for fast dissemination of information, leading to *viral* posts, which are a means to identify trending topics and trendsetters [19]. Retweeting also implies that the user is interested in the content of the original post, and that they are endorsing it, which can be a genuine resource for community detection [3].

A. Top-most retweeted posts

A first kind of analysis we are interested in, is to see whether the two sampling policies differ in terms of the information that they return, with respect to retweets. Towards that end, we conducted the following experiment: We extract the top-k most retweeted posts, that appear in our data. Among other information, Twitter provides the number of times that a post has been retweeted, which serves as the ground truth for ranking. For each of the top-k tweets, we also maintain the number of times that they appear in our dataset.

At the end of this analysis, we obtain a top-k list for each sample, ranked in descending order of their retweet count (ground truth). We want to compare the degree of agreement between the two lists, one obtained from each sample. Given



(a) By iteration (b) By interval size (c) Times Received VS Times Retweeted

Fig. 5. Comparing the top-10000 most retweeted items

that these are ranked lists, we compare them using Kendall's $\tau - b$, which is given by the following equation:

$$\tau = \frac{(n_c - n_d)}{\sqrt{N_1 * N_2}}$$

Kendall's method performs a pairwise comparison of the first n items in the lists, and finds how many pairs appear in the same order (n_c), and how many are not (n_d). In the denominator N_1 and N_2 are the number of items not-tied in the lists. Items which appear in one list but not the other are appended at the end [8]. The result is in the range $[-1, 1]$, where -1 means that the two lists are completely reversed, and 1 is that the two lists are identical. We repeat the comparison for various (sub)list sizes, to understand whether the two lists differ after some point.

We extracted the top-10000 most retweeted items, as they appeared in the 1% and 10% samples. We then compare the two sublists (one for each sample), starting with the top-10 and increasing each time its size by an order of magnitude (top-100, top-1000, etc.). We also compare how many of the most retweeted posts are shared in the two lists. To check whether our finding are biased by Twitter's sampling policy, we also randomly split each sample in half, and rerun our experiment.

Figures 5a-5b show the results of this experiment. A label with S1 and S10 refers to the default or Gardenhose sample, respectively. Labels with P1 or P2 refer to either half of that stream, e.g., S1P1 means the first half of the 1% sample. The omission of a P{1,2} part refer to the entire stream.

Firstly, we observe that up to the top-100 items, the two lists are identical: they contain exactly the same tweets, uniquely identified by their id, (Figure 5b), and they have the exact same ranking (Figure 5a). In other words, if one is only interested in extremely popular tweets, which are but a small fraction of retweeted posts, the 1% sample is adequate. However, if one wants to see the bigger picture, and go beyond the first top-100, the 1% sample starts being problematic.

More specifically, correlation drops to 0.9 when we consider the 10K most retweeted posts. Note that 10K tweets are a very small subset, compared with the entire dataset. As a measure of comparison, the 1% sample returns more than 100K tweets per hour. Despite the high correlation at top-1K and top-10K, it is clear that the 1% sample results in reduced quality, as more items are considered. It is important to note that we obtained similar results when using Kendall $\tau - a$, which only considers common items. Therefore, the drop in correlation is not only due to dissimilar sets. The ranking is affected because the 1% sampling policy does not obtain medium-sized retweets as frequently as the 10% sample.

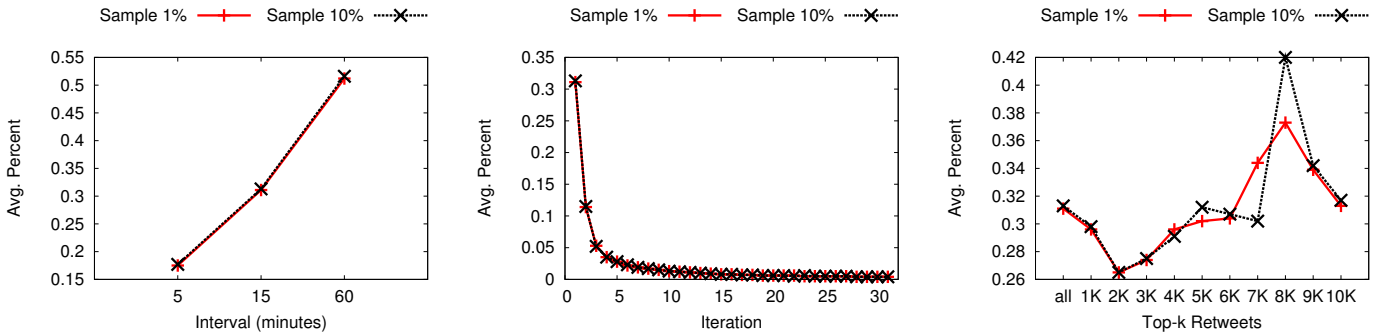
Regarding the halved streams, we observe that the two Gardenhose subsets (S10P1-S10P2) exhibit high correlation, even at the top-10K items. Moreover, the 1% sample essentially shows the same correlation with these two subsets (S1-S10P1, S1-S10P2). This means that the top-most retweeted posts are retrieved multiple times with the 10% sample. On the contrary, this is not the case for the 1% sample (S1-S1P1), validating our claim regarding stale rankings. We expect the correlation to be even lower as we increase the most retweeted items.

As we already described, for each of the tweets appearing in our top-10K most retweeted posts, we maintained the number of times it appears in our dataset. We are then able to rank these items (in descending order), according to the number of times that we encountered them, and compare them against the lists ranked by the actual retweet count, given by Twitter. Figure 5(c) shows the result of this experiment.

Interestingly, the top-1 most retweeted post is not the one that we obtain most times, irrespectively of the sample used. On the other hand, we obtain high correlation starting from the top-5. In the long run, the 10% results in a 0.8 correlation between the two lists, whereas, the 1% sample is significantly lower at 0.7. This practically tells us that the 10% returns items at a much higher rate than the 1%. In combination with the plots in Figures 5a-5b, we conclude that the 1% easily results in stale information.

B. Retweet Burstiness

Viral posts become popular, i.e., they receive a lot of retweets, over a short period of time. The rate at which users retweet information plays an important role in capturing this as an ongoing trending topic. Moreover, a post that rapidly gains attention could be the result of an ongoing event. For this reason, we want to evaluate whether there is a difference between the rates of receiving retweets.



(a) By interval size (b) By iteration, 15' interval (c) By popularity, 1st interval, 15min

Fig. 6. Burstiness of retweeting information

To answer this question, we performed the following experiment: For each of the top-10K most retweeted posts which we extracted from our dataset, we computed how many times we received it within a time period after the tweet was first posted. For instance, with a 5 minute interval, we want to know how many times we received a tweet 5', 10', 15', 20' etc., after it was originally posted. Figures 6(a)-6(c) show these results.

Figure 7(b) shows the following: For each of the top-10K most retweeted posts, we count the percentage of retweets that we received during the first M' minutes after it was posted. Each point in our plot is the average over all of the top-10K most retweeted posts. As expected, when we increase the interval size, more tweets fall within the first interval. It is interesting that more than half of the retweets are received at most within the first hour of the original tweet, while one third of the retweets are received during the first 15 minutes. There is no significant difference between the two sampling policies in that respect.

Figure 6(b) shows how the average of received retweets behaves as a function of the i-th interval, after the original post, with a 15' interval size. As we have seen, during the first 15 minutes, we receive approximately one third of all retweets. This value drops to $\sim$12% in the second quarter and to 5% within 3 quarters of an hour. After this point, we receive very few tweets in every interval. Once more, we do not observe any notable differences between the 1% and 10% samples.

Until now, we averaged over all of the top-most retweeted items. As we saw in our earlier experiment, the behavior was different, if we consider lower ranked items. To check whether this holds for burstiness as well, we did the following: We fix the interval size to 15 minutes and zoom in on the first interval. We split the top most retweeted posts to 1K batches, and rerun our previous experiment (computing the average percentage). For instance, "3K" on the x-axis means that we compute the average of the tweets ranked in positions 2001-3000.

A striking result is that the low-ranked retweeted posts receive (in percentage) more retweets during the first interval. The two samples also differ in these lower ranked retweeted posts, with the most notable differences appearing between the items ranked in positions $[6001, 8000]$. Moreover, posts ranked between $[3001, 7000]$ are closer to the total average. A similar result has been observed with the 2nd interval after the post.

VII. Graph Evolution

One of the major assets of any social networking site, such as Facebook, Twitter, Google+, etc, is its social component. Although, the term "social component" is typically perceived as synonymous to the explicit social graph, there is more information which can be used in that direction.

More specifically, users engage in discussions, reply to each other either to form an arguement or respond to questions, endorse views by favoriting, "liking", and retweeting, or simply mention other entities / users in the content they upload. All such actions are explicit forms of interaction between the users. In that sense, the social graph is a subset of what constitutes the social component of the social medium.

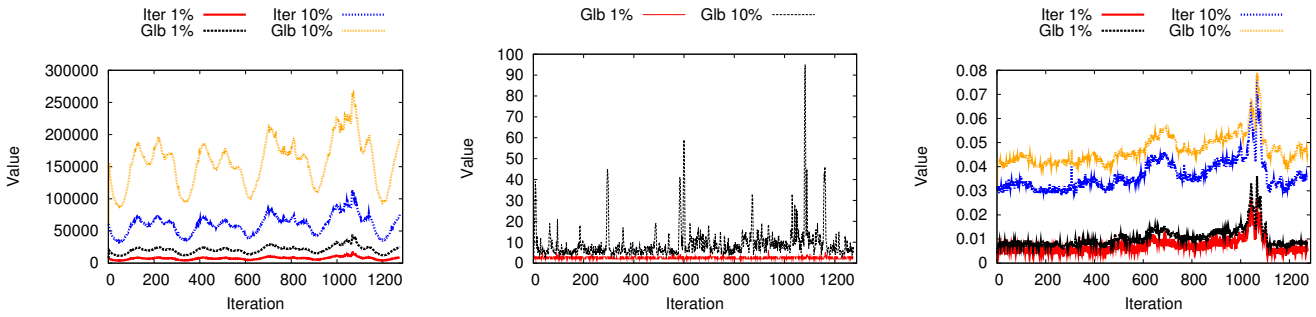
We are interested in identifying key properties of the retweet graph, extracted over time from the incoming stream of tweets. We would like to know how well these properties correlate with the ground truth data, as presented in [17] where the entire Twitter stream was used, when we consider the 10% sample.

A. Temporal Retweet Graph

Retweets are a very particular characteristic of the Twitter service. As already mentioned, it allows users to repost tweets, thereby endorsing and acknowledging the original poster at the same time. If user $\mathcal{A}$ retweeted a post, originally posted by user $\mathcal{B}$, then we add an edge from user $\mathcal{A}$ to user $\mathcal{B}$. This is a directed graph, much like Twitter's explicit social graph. Note that we do not focus on a particular tweet in this case, as this would form a star-shaped graph. Therefore, the graph can be the result of multiple individual tweets, posted at different timestamps.

Compared with [17], we want to see how the retweet graph changes over time. Rather than taking daily snapshots of the graph and average them, we would like our graph to incorporate a more continuous notion of time. To achieve this, the edges of our graph are weighted and we decay them over time. The edges are removed when their weight drops below a certain threshold. More specifically, we construct our retweet graph in the following manner:

- **Step 1:** We use an interval size, similar to the one used for the Retweet analysis. We extract the retweet graph



(a) Size (b) Maximal Connected Component Size (c) Clustering Coefficient (overall)

Fig. 7. Statistical properties of the extracted retweet graph, over time

using the tweets that were posted during the first interval. This is our starting graph $\mathcal{G}_0$.

- **Step 2:** Proceed to the next (i-th) interval. Extract the graph of that interval, which we denote by $\mathcal{G}_i$
- **Step 3:** The initial edge weight from a node $\mathcal{X}$ to a node $\mathcal{Y}$ is equal to the number of times that user $\mathcal{X}$ retweeted any post from node $\mathcal{Y}$. We normalize the weights, so that, for each node, the total outgoing edge weight is 1.
- **Step 4:** Decay the graph $\mathcal{G}_{i-1}$, that we have aggregated until this point, using an exponential function. This means that the weight of each edge becomes $w = w * \exp^{-x}$. If that edge drops below a certain threshold t, remove the edge. This implies that the edge is too old, and has not been updated recently.
- **Step 5:** Add the decayed graph $\mathcal{G}_{i-1}$ to the one extracted at the current iteration, $\mathcal{G}_i$. The graph contains the union of the two node sets. We add an edge between two nodes, iff there is such an edge in $\mathcal{G}_i$ or $\mathcal{G}_{i-1}$. If such an edge exists in both graphs, the edge weight is the sum of the two individual weights in either graph. Proceed with Step 2, until all intervals have been processed.

Figure 7 shows the results of this experiment. In particular, it depicts the number of nodes that the entire graph contains. We have plotted both the statistics for the aggregated graph until the i-th iteration, as well as the statistics for graph $\mathcal{G}_i$ of each iteration. We note that the graph exhibits a periodicity akin to the one shown in the data volume and sentiment analysis. As we can see, the global graph contains the most nodes of all cases. However, its size does not necessarily increase, as old nodes are discarded, because they did not appear in a more recent interval.

Figure 7b shows the size of the Largest Connected Component (LCC), as a function of the interval. It is interesting, that the size of the LCC does not share the same periodicity we have seen in other cases. Rather, in various occasions, the graph will increase its size significantly, and then return to normal values. Finally, we have computed the clustering coefficient of the 4 graphs we extract, 2 for the global case and 2 for each iteration 1 for each sample). It is interesting that, over time (Figure 7c), the retweet graph extracted from the Gardenhose sample has a clustering coefficient very close to the one reported by [17]. This, in fact, means that the retweet graph we extract from the Gardenhose, yields similar results to the ground truth data.

VIII. Linguistic Analysis

As a final experiment, we would like to see whether there is a correlation between the spoken languages in Twitter, and the ground truth obtained from studies in the physical world. Moreover, we want to check whether there are any differences regarding the two sampling policies. To perform this experiment, we used language detection software [3] and obtained ground truth information from Wikipedia[4,5]. We map each tweet to a language and count the number of tweets with that language. We then derive a ranked list for the languages, based on the absolute counts, and we compare the derived list for each sample with the ground truth using Kendall τ.

Table III depicts the results of this comparison. Correlation is lower when we consider native speakers, as opposed to lists ranked by the number of people in the world who speak that language. Regardless, even if we account for the fact that the language detection software is not perfect (i.e., is not 100% accurate), the correlation between the extracted list from Twitter data and the ground truth is extremely low. This holds for both sample sizes. Therefore, there is an inherent bias in the data, which is not due to the sampling policy, but mostly because of the user base of the service itself. This means that researchers on the field of linguistic analysis, who rely on Twitter data, should be weary of this inherent bias.

TABLE III. Comparison of languages extracted from samples

	Ethnologue	**Spoken Popularity**
Sample 1%	0.158	0.342
Sample 10%	0.155	0.342

IX. Efficiency

Table IV shows the efficiency of each experiment for the two sample sizes. In particular, we measure the wall clock

[3] https://code.google.com/p/language-detection/
[4] http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers
[5] http://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers

time from the point that we started processing the input, up to the point that the full output was written (either to the standard output, or to a file). Our experiments were run on a single Quad-core machine @3.4GHz, with 16Gb RAM, running Ubuntu Linux.

Despite the fact that the actual data differ by an order of magnitude, the running times do not differ by the same amount. There are, of course, various reasons for that, including caching, other processes (e.g., daemons) running simultaneously, process context switching, etc. It is clear, however, that processing takes substantially more time, on a single machine.

TABLE IV. EFFICIENCY OF EXPERIMENTS (SECONDS)

	Sample 1%	**Sample 10%**
Sentiment Analysis	147.276	2058.264
RT Graph Evolution	175.061	2531.362

X. CONCLUSIONS AND FUTURE WORK

Twitter provides various ways to access its data, which results in different trade-offs. In this paper we considered the problem of evaluating the differences between the default sampling policy of Twitter and the Gardenhose. We compared the two policies on various levels, including spatial characteristics, properties of the retweet graph, as well as more analytical results on spoken languages and sentiment analysis. Our analysis also had a strong temporal focus, and we showed that less popular retweets are better captured by the Gardenhose. Moreover, a temporal retweet graph can obtain properties significantly similar to the ground truth.

As future work we plan to investigate the impact of the two samples on other analytical processes such as event detection and term co-occurrence, and focus on other types of social interaction, such as replies, and topical communities.

ACKNOWLEDGMENTS

This work has been co-financed by EU and Greek National funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Programs: Heraclitus II fellowship, THALIS - GeomComp, THALIS - DISFER, ARISTEIA - MMD and the EU funded project INSIGHT.

REFERENCES

[1] Amr Ahmed, Liangjie Hong, and Alexander J. Smola. Hierarchical geographical modeling of user locations from social media posts. WWW, 2013.

[2] Gennady L. Andrienko, Natalia V. Andrienko, Harald Bosch, Thomas Ertl, Georg Fuchs, Piotr Jankowski, and Dennis Thom. Thematic patterns in georeferenced tweets through space-time visual analytics. *Computing in Science and Engineering*, 15(3):72–82, 2013.

[3] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Inuence-based network-oblivious community detection. In *ICDM*, 2013.

[4] Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. WSDM, 2010.

[5] Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. In *ACL-HLT*, 2011.

[6] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. Political polarization on twitter. In *ICWSM*, 2011.

[7] Andrew Crooks, Arie Croitoru, Anthony Stefanidis, and Jacek Radzikowski. #earthquake: Twitter as a distributed sensor system. *T. GIS*, 17(1):124–147, 2013.

[8] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing partial rankings. *SIAM J. Discret. Math.*, 20(3):628–648, 2006.

[9] Saptarshi Ghosh, Muhammad Bilal Zafar, Parantapa Bhattacharya, Naveen Sharma, Niloy Ganguly, and Krishna Gummadi. On sampling the wisdom of crowds: Random vs expert sampling of the twitter stream. In *CIKM*, 2013.

[10] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. In *INFOCOM*, pages 2498–2506, 2010.

[11] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the twitter stream. WWW, 2012.

[12] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, 2004.

[13] Yuheng Hu, Fei Wang, and Subbarao Kambhampati. Listening to the crowd: Automated analysis of events via aggregated twitter sentiment. In *IJCAI*, 2013.

[14] Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. Emoticon smoothed language models for twitter sentiment analysis. In *AAAI*, 2012.

[15] Michael Mathioudakis and Nick Koudas. Twittermonitor: trend detection over the twitter stream. In *SIGMOD*, 2010.

[16] Yelena Mejova, Padmini Srinivasan, and Bob Boynton. Gop primary season on twitter: "popular" political sentiment in social media. In *Proceedings of the sixth ACM international conference on Web search and data mining*, WSDM '13, pages 517–526, 2013.

[17] Fred Morstatter, J ürgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. In *ICWSM*, 2013.

[18] Georgios Paltoglou and Mike Thelwall. Twitter, myspace, digg: Unsupervised sentiment analysis in social media. *ACM Trans. Intell. Syst. Technol.*, 3(4):66:1–66:19, September 2012.

[19] Diego Saez-Trumper, Giovanni Comarela, Virgílio Almeida, Ricardo Baeza-Yates, and Fabrício Benevenuto. Finding trendsetters in information networks. In *SIGKDD*, pages 1014–1022, 2012.

[20] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, 2010.

[21] Anthony Stefanidis, Andrew Crooks, and Jacek Radzikowski. Harvesting ambient geospatial information from social media feeds. *GeoJournal*, 78(2):319–338, 2013.

[22] J. Sutton, Leysia Palen, and Irina Shlovski. Back-channels on the front lines: Emerging use of social media in the 2007 southern california wildfires. 2008.

[23] George Valkanas and Dimitrios Gunopulos. Location extraction from social networks with commodity software and online data. In *ICDM Workshops (SSTDM)*, 2012.

[24] George Valkanas and Dimitrios Gunopulos. How the live web feels about events. In *CIKM*, 2013.

[25] George Valkanas, Dimitrios Gunopulos, Ioannis Boutsis, and Vana Kalogeraki. An architecture for detecting events in real-time using massive heterogeneous data sources. In *BigMine*, pages 103–109, 2013.

[26] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *ICWSM*, 2011.