Rhea: Adaptively Sampling Authoritative Content from Social Activity Streams

Panagiotis Liakos*, Alexandros Ntoulas†‡ and Alex Delis*§
*University of Athens, Athens, Greece, Email: {p.liakos, ad}@di.uoa.gr
†LinkedIn, Mountain View, CA, Email: ntoulas@gmail.com
§New York University Abu Dhabi, Abu Dhabi, U.A.E.

***Abstract*—Processing the full activity stream of a social network in real time is oftentimes prohibitive in terms of both storage and computational cost. One way to work around this problem is to take a sample of the social activity and use this sample to feed into applications such as content recommendation, opinion mining, or sentiment analysis. In this paper, we study the problem of extracting samples of *authoritative* content from a social activity stream. Specifically, we propose an adaptive stream sampling approach, termed `Rhea`, that processes a stream of social activity in real-time and samples the content of users that are *more likely* to provide *influential* information. To the best of our knowledge, `Rhea` is the first algorithm that dynamically adapts over time to account for evolving trends in the activity stream. Thus, we are able to capture high quality content from emerging users that contemporary white-list based methods ignore. We evaluate `Rhea` using two popular social networks reaching up to half a billion posts. Our results show that we significantly outperform previously proposed methods in terms of both recall and precision, while also offering remarkably more accurate ranking.**

I. Introduction

The tremendous scale of content generation in online social networks brings several challenges to applications such as content recommendation, opinion mining, sentiment analysis, or emerging news detection, all of which have an inherent need to mine this content in real time. As an example, the daily volume of new *tweets* posted by users of `Twitter` surpasses 500 million.[1] However, not all generated online social activity is useful or interesting to all applications. Using `Twitter` again as an example, more than 90% of its posts is actually conversational and of interest strictly limited to a handful of users, or spam [10]. Therefore, applications such as emerging news detection that operate on the entire stream, spend a lot of computational cycles as well as storage in processing posts that are not very useful.

One way to solve this problem is, instead of processing the social activity stream in its entirety, to take a sample of the activity and operate on the sample. Through sampling, our goal is to still capture the important and interesting parts of the activity stream, while reducing the amount of data that we would have to process. To this end, one obvious approach is to perform random sampling, i.e., randomly pick a subset of the activity stream and use that in the respective application. A more effective approach however, is to sample content published in the activity stream only from the users that are considered authoritative (or *authorities*).[2] By sampling the posts of authoritative users from the stream, we are reportedly [24] more likely to produce samples that are of *high-quality*, with limited conversational content and less spam.

The challenge in sampling high quality content from a social activity stream lies therefore in identifying authoritative users. Existing work deploys white-lists of users that are likely to produce authoritative content [9], [10], [20], [24] and samples their activity. Although such approaches have been shown to work well for certain applications, we will show experimentally that they are unable to cope with the dynamic nature of a social activity stream where, for example, new users emerge as authorities and old ones fade out. Other prior efforts on identifying authoritative users in social networks (not streams) have focused on computing a relative ranking of users based on network attributes [2], [4], [5], [11], [16], [25]. We build on the findings of such approaches to identify authorities likely to produce useful content; our approach is different however, as we cannot presume that the complete structure of the social network is available, nor that we can afford to process the network offline.

We operate with the more practical assumption that we have incomplete access to the social network. In other words, we do not know which users exist in the network but we simply observe some partial activity from a social activity stream. Our goal is to produce high quality samples from such streams that will still be as useful as possible compared to being able to access the entirety of the social network and the activity within.

We propose `Rhea`,[3] an adaptive algorithm for sampling authoritative social activity content. `Rhea` forms a *network of authorities* as it processes a stream and includes in its sample only the content published by the top-K authorities in this network. Given a social activity stream with user interactions (e.g., answers in `Q&A` sites or mentions in the case of `Twitter`) we create a weighted graph used to quantify user authoritativeness. To deal with the potentially enormous amount of items that we encounter in the stream and limit memory blowup, we construct a highly compact, yet extremely efficient sketch-based novel data structure to maintain the authoritative users of the network. Our experimental results with half a billion posts from two popular social networks show significant improvements with regard to various binary

This work has been partially supported by the University of Athens Special Account of Research Grants № 13233.

‡This work was done before author joined LinkedIn.

[1] http://www.internetlivestats.com/twitter-statistics/

[2] We use terms *authoritative users* and *authorities* interchangeably.

[3] Rhea was the Titaness daughter of the earth goddess Gaia and the sky god Uranus. Her name stands for "she who flows".

and ranked retrieval measures over previous approaches. Rhea is able to sample significantly more *relevant* documents, with *higher precision* and remarkably more accurate *ranking* compared to sampling based on static white-lists of authoritative users. Our approach is generic and can be used with any online social activity stream, as long as we can observe indicators of authoritativeness in the stream.

In summary, we make the following contributions:

1) We propose Rhea, a stream sampling algorithm, that employs network-based measures to dynamically elicit authoritative content of social activity. To the best of our knowledge, this is the first work that addresses the problem of dynamically sampling the posts of authoritative users from a social activity stream.
2) We evaluate Rhea with datasets reaching up to half a billion posts from two popular social networks and show that it outperforms contemporary approaches with regard to precision, recall, and ranking accuracy.
3) We empirically demonstrate that static white-lists cannot always capture *temporal* changes in rankings of authorities, and thus, are not an appropriate choice when sampling authoritative content from streams.

II. Identifying Authorities in Streams

A. Network of Authorities from Social Activity

Streams of social activity reveal very little about the respective network structure. Depending on the social network, users may perform certain actions like "posting" messages or "liking" content other users have posted. For example in Twitter, tweets may mention another user's *@username*, in Facebook users may tag another user, in Linkedin users can make endorsements, while in Q&A sites such as StackOverflow, users can provide answers to other users' questions. The aforementioned actions (mentions, endorsements, answers, etc.) as well as their direction may often be considered as indications of importance, and can be used to form a network of authorities from the respective stream. More specifically, users receiving numerous mentions or regularly providing answers, without reciprocating these actions with the same frequency, may be deemed as *important* in the network [22].

To illustrate the process of deriving a network of authorities from social activity, we provide an example of a stream featuring the three tweets depicted in Figure 1. In the first element of the stream, *@user1* creates a mention to user *@SLAM* by retweeting a post of that user regarding the injury of a basketball player. Then, the same user retweets some additional information on the same story from the same source. These posts appear in the *feeds* of the users that follow *@user1*. Soon, *@user2* posts a reply to *@user1* and reports that another source (*@SI*) has also confirmed the story. Overall, there are 4 mentions in this stream, most of which offer valuable *evidence* regarding user importance. However, one of the mentions (to *@user1*) is actually only a reply; the respective tweet is conversational and the user simply intends to notify another user. Similarly, in a Q&A site, providing answers is usually an indication of authoritativeness, even though some answers may be inaccurate.

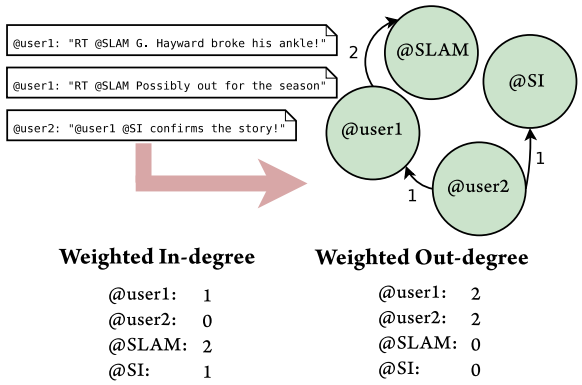


Fig. 1: Deriving a network of authorities from a social activity stream. Potential authorities may be identified by applying measures on the resulting weighted directed graph.

Figure 1 also depicts the actual process of forming a network of authorities out of this particular social activity stream. The network is represented as a directed weighted graph. For each mention in the stream we create an edge from the source node (i.e., user) to the receiving node. If the edge is already present, we increase the respective weight by 1. Using the 3 tweets of our example we can detect a total of 4 nodes. We observe that one of the nodes stands out with regard to weighted in-degree (*@SLAM*). However, we also see that based on weighted in-degree alone, we cannot differentiate between receiving mentions indicating importance and replies. To this end, we can additionally utilize the weighted out-degree to quantify the extent to which these actions are reciprocated, as we discuss next.

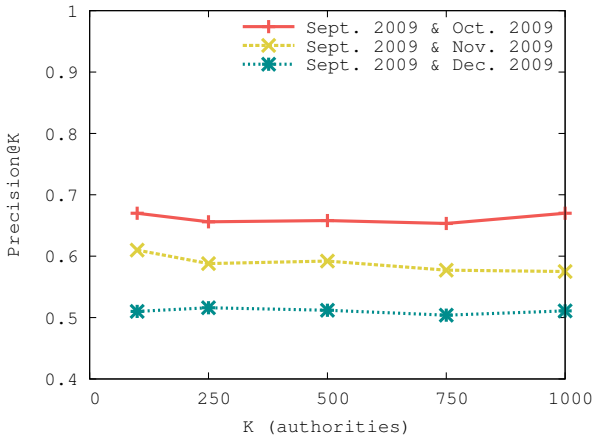
B. Ranking the Authorities

Numerous prior efforts have utilized network structure to identify authorities and exploit the content they produce [2], [4], [11], [25]. Although in our setting we cannot recover the complete network structure, there are usually indications of expertise inherent in the social activity stream that we can utilize. When a user mentions another user in Twitter she is either acknowledging the authority of the latter, or trying to engage in a conversation. Both these actions typically imply that the initiating user considers herself less authoritative than the target user. On the other hand, receiving a mention is often an indicator of importance for the recipient. Similarly, asking questions in Q&A sites is usually a negative indicator of authoritativeness, whereas providing answers is a positive one. Therefore, one way to capture this balance is to compute the fraction of the difference between these indicators of importance.

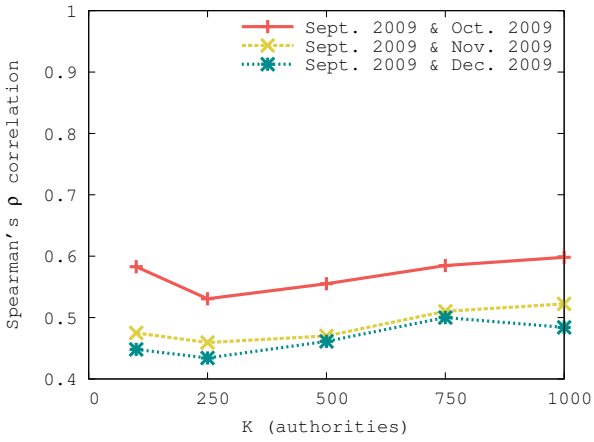
Zhang et al. [25] focus on Q & A communities and propose *z-score*, a measure that builds on positive and negative predictors of *expertise*. The *z-score* of user u is formally defined as:

$$z(u) = \frac{a(u) - q(u)}{\sqrt{a(u) + q(u)}} \quad (1)$$

where, $a(u)$ is the number of questions u has answered and $q(u)$ is the number of questions u has asked. Through crowdsourcing Zhang et al. show that *z-score* outperforms measures



(a) Precision@K results.



(b) Spearman's ρ results.

Fig. 2: Precision@K (a) and Spearman's ρ (b) results for the authorities extracted from the tweets of September 2009, using the rankings resulting from the tweets of the three subsequent months. Both metrics reveal that the correlation between rankings of authorities according to the tweets of subsequent months weakens significantly with time.

such as the *in-degree* as well as sophisticated approaches based on *PageRank* [13] and *HITS* [12] when identifying distinguished users in social networks. We build on this finding and propose *auth-value*, a generalized version of *z-score* for a wide range of social networks, that we formally define as:

$$auth(u) = \frac{in(u) - out(u)}{\sqrt{in(u) + out(u)}} \quad (2)$$

where, $in(u)$ is the weighted in-degree of u in the network of authorities and $out(u)$ is her respective weighted out-degree. Thus, our *auth-value* measure enables us to extract the authoritative users of a network in which social activity does not necessarily imply user *expertise*. As the effectiveness of *z-score* against other measures has been previously exhibited [25], we rely on Eq. (2) to measure authoritativeness and our focus is on applying it effectively in a streaming setting.

Taking into account both positive and negative predictors of importance through Eq. (2) allows us to differentiate between authorities and frequent posters. In particular, users who are frequently mentioned in conversational tweets or provide (possibly incorrect) answers to numerous questions, are also expected to make a lot of mentions to other users or frequently ask questions, and will be penalized by Eq. (2) for doing so. More specifically, such users are expected to a exhibit an *auth-value* that is negative or close to zero. In contrast, authoritative users who receive much more mentions than they give or answer significantly more questions than they ask will exhibit high *auth-values*. We note, that $auth(u)$ is susceptible to spam-farms that may attempt to boost the values of certain users; however, this is the case with alternative network measures as well, e.g., *in-degree*, *PageRank*, or *HITS*. Thus, we consider that fighting web-spamming is beyond the scope of our work. Moreover, we use the notion of authoritativeness to describe influential contributors of a network *regardless* of the diversity of topics discussed. Our focus is on the entire activity and thus, our goal is to distinguish the highly influential players *overall*, as the case is with prior stream sampling efforts [10].

TABLE I: Top-10 authorities for the tweets of 3 months.

	October 2009		November 2009		December 2009	
	user u	auth(u)	user u	auth(u)	user u	auth(u)
1	justinbieber	393.885	justinbieber	448.815	justinbieber	433.185
2	donniewahlberg	358.286	donniewahlberg	249.988	nickjonas	249.558
3	tweetmeme	263.103	revrunwisdom	242.807	revrunwisdom	222.571
4	revrunwisdom	237.964	tweetmeme	195.379	donniewahlberg	202.996
5	mashable	229.650	addthis	186.282	tweetmeme	183.603
6	addthis	212.325	**ddlovato**	181.720	jonasbrothers	182.882
7	**ddlovato**	204.910	**luansantanaevc**	167.514	addthis	181.403
8	jordanknight	191.045	jordanknight	167.197	omgfacts	154.136
9	jonasbrothers	175.054	jonasbrothers	165.520	mashable	153.616
10	lilduval	174.616	mashable	164.496	johncmayer	147.241

C. Limitations of Static Lists of Authorities

Previous approaches on sampling the activity of authoritative users from social streams employ *white-lists* of authorities extracted from user annotated content [9], [10], [20], [24]. In particular, social networks often enable users to create *lists* that group together distinguished users. We can form a white-list of authorities by including users with considerable appearances in such user-generated lists [9]. Although this approach can work well in some cases, static white-lists may often be outdated, featuring inactive accounts or users that are no longer receiving attention. Activity in social networks is highly *dynamic* and authorities tend to *rise and fall* with time. To quantify how dynamic social activity is, we used `Twitter` posts from 3 consecutive months. We created a white-list for each month, that comprises the most authoritative users according to their *auth-values*, and examined the similarity of these white-lists.

Table I shows the top-10 authorities for these 3 months. We observe that even at the first 10 positions user rankings vary across different months. For instance, user *ddlovato*[4] started from the 7^{th} spot in October, moved up to the 6^{th} spot in November, before dropping off from the list in December.

[4] *ddlovato* is the account of American singer/actress Demi Lovato: https://twitter.com/ddlovato.

This is an example of a user that receives increasing attention over time before eventually being surpassed by other emerging users later on. Similarly, user *luansantanaevc*[5] appeared in the first 10 positions only for the tweets of November. This is an example of a user that received attention *temporarily*. More importantly, this user started posting tweets using a second account (*luansantana*) on January 19^{th}, 2011, without deactivating the first account. Hence, we observe that white-lists can be *unstable* and quickly become *out-of-date*.

To further quantify the volatility of rankings in white-lists, we examined their similarity over time based on the percentage of the users appearing in the list of September 2009 that also appeared in the lists of the 3 subsequent months. More specifically, we consider the ranking resulting from the tweets of September to be the *ground-truth*, and calculate the *Precision@K* achieved in the following 3 months. Figure 2(a) depicts the *Presicion@K* results for $100 \leq K \leq 1{,}000$ for October, November, and December, respectively. For October (the immediately following month), less than 70% of the authorities of September are also identified as authorities, for all values of K examined. As expected, *Precision@K* deteriorates very quickly in the following months. For the same range of K, we get that $0.57 <$ Precision@K ≤ 0.61 for November, and $0.50 <$ Precision@K ≤ 0.51 for December. Overall, *Precision@K* remains relatively stable as we increase K and deteriorates as we increase the time interval from the ground truth. We additionally measure the rank correlation using Spearman's ρ with the similar setting of considering September as the ground truth. Figure 2(b) illustrates Spearman's ρ results for the same pairs of months. There is moderate correlation between the rankings of users, which remains stable as we increase K. However, the rank correlation also weakens as we increase the time interval from the ground-truth.

Our findings strongly suggest that white-lists are inappropriate when extracting authoritative users from streams, as social activity is dynamic; instead, we need an adaptive algorithm. We proceed by presenting such an algorithm and measuring its effectiveness against contemporary white-list based approaches.

III. RHEA: STREAM SAMPLING FOR AUTHORITATIVE CONTENT

In this section we present `Rhea`, an adaptive sampling algorithm for authoritative content from social activity streams. More formally, `Rhea` seeks to produce a sample $\hat{S}$ of a stream S such that $\forall s \in S$ whose respective user is in the *top-K* authorities of the network according to Eq. (2), $s \in \hat{S}$.

This endeavor involves three main challenges: 1) Online social networks are ever increasing and users publishing content may surpass 1 billion [6]. Hence, maintaining user information as we process the stream may be costly in both memory requirements and computational time. 2) Ranking users according to their authoritativeness and classifying their content as relevant or non-relevant, often requires reckoning in multiple measures. 3) Finally, many elements we opt to include in the sample as we process the stream may actually be published by non-authorities. Thus, we need to filter out posts that mistakenly lurked in our sample. In this section, we discuss the individual pieces of `Rhea` that address these three challenges and then present our algorithm.

[5] *luansantanaevc* is the old account of Brazilian singer Luan Santana: https://twitter.com/luansantanaevc.

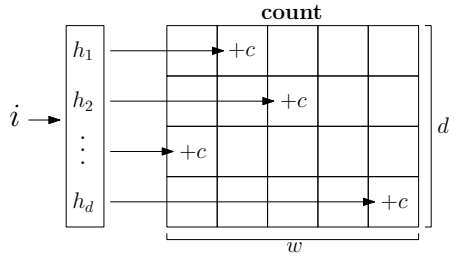


Fig. 3: COUNT-MIN Sketch update process.

A. Maintaining User Information

*1) **Frequent Items**:* `Rhea` maintains a *limited view* of the social network based on the social activity stream. In particular, `Rhea` is aware of the weighted in- and out-degrees of each user in the stream, as depicted in the weighted directed graph of Figure 1. In practice, we expect that an enormous number of users will participate in the activity stream of an online social network. Efficiently mapping their respective weighted in- and out- degrees with structures such as hash tables would require memory that far surpasses that of a modern day computer. Moreover, resizing such hash tables would be necessary to maintain new users encountered in the stream, and would eventually cause serious bottlenecks in terms of CPU cycles.

The COUNT-MIN sketch [7] is a well-known and widely-used [1], [18] sublinear space data structure for the representation of high-dimensional vectors. COUNT-MIN sketches allow fundamental queries to be answered efficiently and with strong accuracy guarantees. It is particularly useful for summarizing data streams as it is capable of handling updates at high rates. The sketch uses a two-dimensional array of w columns and d rows, where $w = \lceil \frac{e}{\epsilon} \rceil$, $d = \lceil \frac{ln(1)}{\delta} \rceil$, and the error in answering a query is within a factor of ϵ with probability δ. A total of d pairwise independent hash functions is also used, each one associated with a row of the array. Figure 3 illustrates the update process of a COUNT-MIN sketch for our specific problem. Consider that an update (i, c) arrives, indicating that user's i count should be incremented by c. The array $count$ is updated as follows: for each row j of $count$ we apply the corresponding hash function to obtain a column index $k = h_j(i)$ and increment the value in row j, column k of the array by c, i.e., $count[j, k] += c$. This allows for retrieving at any time an (over)estimation of the count of an event i using the least value in the array for i, i.e., $\hat{a_i} = min_j count[j, h_j(i)]$.

`Rhea` keeps track of both positive and negative indicators of importance. Thus, we employ two COUNT-MIN sketches to compactly maintain both these indicators for all users appearing in a stream.

*2) **Reducing the Processing Overhead through Sampling**:* Palguna et al. [17] show that a *uniform random sample with replacement* of enough size is able to guarantee with strong accuracy that i) elements that occur with frequency more than θ in the stream occur with frequency more than $(1 - \frac{\epsilon}{2})\theta$ in the sample and ii) elements that occur with frequency less than

Algorithm 1: put($Top\text{-}K\text{-}Heap, key, value$)

```
input  : A Top-K-Heap structure and a key associated with a
         value to be inserted in the Top-K-Heap.
output : The updated Top-K-Heap.
1  begin
2      if Top-K-Heap.size() < K then
3          if Top-K-Heap.contains(key) then
4              Top-K-Heap.replace(key, value);
5          else
6              Top-K-Heap.push(key, value);
7      else
8          if Top-K-Heap.contains(key) then
9              Top-K-Heap.replace(key, value);
10         else if value > Top-K-Heap.low() then
11             Top-K-Heap.pop();
12             Top-K-Heap.push(key, value);
13     return Top-K-Heap;
```

$(1-\epsilon)\theta$ in the stream occur with frequency less than $(1-\frac{\epsilon}{2})\theta$ in the sample, where $\theta \in [0,1]$ and $\epsilon \in [0,1]$. In addition, they experimentally show that the behavior of the *Bernoulli sampling scheme* is very similar and primarily influenced by the sample size alone. Obviously, we are unable to use *uniform random sample with replacement*, as elements of the stream are only seen once. However, we can employ the *Bernoulli sampling scheme*. In particular, we can include each element of the stream in our authorities' network formation process with probability p and exclude the element with probability $1-p$, independently of other elements, where $p \in (0,1]$. This allows us to reduce the computational overhead of `Rhea` without sacrificing its effectiveness. We thoroughly investigate the impact of p in our evaluation to come up with the size of the sample that will facilitate our set requirements.

B. Ranking Authorities

COUNT-MIN sketches provide answers to point and dot product queries with strong accuracy guarantees. Using two such sketches, we are able to approximate the number of positive and negative indicators of importance a user exhibits. This is enough to provide us with an approximation of a user's *auth-value* through Eq. (2). However, we are not interested in the *absolute* value of $auth(u)$. Rather, we wish to know at any time whether a user's value is among the top-K overall. To this end, we employ a structure we term $Top\text{-}K\text{-}Heap$ to hold user elements with associated *auth-values*. A $Top\text{-}K\text{-}Heap$ *puts* an element in the structure if its value is larger than the minimum value currently on the structure or the structure holds less than K elements. In case the element is already inserted, we update its value accordingly; otherwise, we first remove the element with the smallest value. Thus, a $Top\text{-}K\text{-}Heap$ holds a maximum of K elements. In addition, duplicate values are allowed, as users may exhibit the same *auth-value*.

A *min-heap* [3] allows for duplicate values and enables us to examine the minimum element of our structure in *constant* time. In addition, *min-heaps* support insertion of elements or removal of the minimum element in *logarithmic* time. Therefore, if the element is not present in the structure, we can place it in a *min-heap* and remove the root holding the minimum value in *logarithmic* time. However, examining if an element is already in a min-heap takes *linear* time. To alleviate this problem, we additionally employ a *hash-table* to hold the inserted elements, which allows for examining the presence of an element in our $Top\text{-}K\text{-}Heap$ in *constant* time. We note that K is insignificant compared to the total number of users, and the cost of using an additional *hash-table* is negligible.

Algorithm 1 details the insertion in a $Top\text{-}K\text{-}Heap$. Lines 2-6 concern the case when the $Top\text{-}K\text{-}Heap$ holds less than K elements. If the new element is already inserted we replace its value (Line 4), i.e., we remove the old element from the *min-heap* and the *hash-table* and insert the new one with the updated value (O(logn)). If the new element is not in the structure, we simply place it inside (Line 6), i.e., we insert it to both the *min-heap* and the *hash-table* (O(logn)). Lines 7-12 are executed in the case when the $Top\text{-}K\text{-}Heap$ holds exactly K elements. If the latter is true and the element to be added is already inserted, we replace its value as before (Line 9). However, if the new element is not already in the $Top\text{-}K\text{-}Heap$, we examine if its value is larger than the minimum value on the $Top\text{-}K\text{-}Heap$ (Line 10), and remove the root of the $Top\text{-}K\text{-}Heap$ before inserting it. This requires us to access the minimum value (O(1)), remove the root of the *min-heap* (O(logn)) and the respective element in the *hash-table* (O(1)), and then insert the new element in the *min-heap* (O(logn)) and the *hash-table* (O(1)). We note that Eq. (2) may both increase or decrease as elements appear in the stream. Therefore, when we update an element in the $Top\text{-}K\text{-}Heap$ with a value that is smaller than the one previously held, it might be the case that the element should no longer be part of the top-K. However, as we update the $Top\text{-}K\text{-}Heap$ with every element that appears on the stream, the element that would actually belong to the top-K will claim its position at its next appearance, and will be included in the sample.

C. Filtering-out Non-relevant Activity

`Rhea` makes decisions based on what appears to be optimal at the time. During stream processing, `Rhea` may deem as a top-K authority a user that temporarily exhibits a high *auth-value* but is actually not among the top-K overall for the particular stream. Thus, posts of non-authorities may end up in our sample, i.e., we lose in *precision*. Similarly, `Rhea` might stumble upon posts of an authority that is not yet identified as such. This will lead to relevant posts being excluded from our sample, i.e., we lose in *recall*. For this latter case, we are unable to improve our *recall* at a later stage, as the elements that we discard from the stream are lost. However, for the former case we can perform a post-processing step to filter-out non-relevant posts using the more refined classification model that is formed after seeing a good portion of the stream. For each document included in our sample, `Rhea` examines the respective user that published it. If the user is contained in our $Top\text{-}K\text{-}Heap$, we keep the document in the sample; otherwise, we discard it. In our evaluation, we investigate the impact of this technique in detail.

D. The Proposed Rhea Algorithm

Algorithm 2 outlines our proposed `Rhea` method for stream sampling. `Rhea` processes a stream S with elements of social activity. Each element contains some content and is associated with a user and a timestamp. `Rhea` takes as its

Algorithm 2: Rhea(S, K, p)

input : A stream S, a parameter $K > 0$ and a probability $p \in (0, 1]$.
output : A set $\hat{S} \subset S$ containing elements whose respective users are likely to be among the top-K w.r.t. to the *auth-value*.

1 **begin**
2 $Top\text{-}K\text{-}heap \leftarrow \emptyset$;
3 $CMSin \leftarrow \emptyset$;
4 $CMSout \leftarrow \emptyset$;
5 **foreach** $s \in S$ **do**
6 **if** $random(0, 1] < p$ **then**
7 $(in, out) \leftarrow extractIndicators(s.message)$;
8 $CMSin[in] += 1$;
9 $CMSout[out] += 1$;
10 $auth_{user} \leftarrow \frac{CMSin[s.user] - CMSout[s.user]}{\sqrt{CMSin[s.user] + CMSout[s.user]}}$;
11 **if** $auth_{user} > Top\text{-}K\text{-}heap.low()$ **then**
12 $put(Top\text{-}K\text{-}heap, user, auth_{user})$;
13 $\hat{S}.put(s)$;
14 **foreach** $s \in \hat{S}$ **do**
15 **if** $s.user \notin Top\text{-}K\text{-}heap$ **then**
16 $\hat{S}.remove(s)$;
17 **return** $\hat{S}$;

input parameters K and p, that specify the amount of authorities whose activity we wish to sample, and the probability according to which we process an element in the stream to form our network of authorities, respectively. The output is a sample of S containing elements whose respective users are likely among the top-K w.r.t. the *auth-value*.

Rhea begins by initializing the structures to be used while processing the stream (Lines 2-4), i.e., a $Top\text{-}K\text{-}Heap$ to hold the current K users with the highest *auth-value* in the stream, and two COUNT-MIN sketches to maintain the weighted in- and out-degree of each user. Then, we process the elements of the stream (Line 5), a phase that involves two actions:

Creating the Network of Authorities (Lines 6-9): We apply a *Bernoulli sampling scheme* and use an element of the stream with probability $p \in (0, 1]$ to extract positive and negative indicators of importance (Line 6-7). The extracted indicators are used to update the two COUNT-MIN sketches (Lines 8-9).[6] Hence, sketches CMSin and CMSout keep track of the weighted in- and out-degrees of the users of the formed authorities' network, respectively.

Stream Sampling for Authoritative Content (Lines 10-13): First, we derive an approximation of the *auth-value* of the respective user of the current element of the stream (Line 10). Then, we compare with the lowest value in the $Top\text{-}K\text{-}Heap$ to decide whether the current user is an authority, and thus, her activity must be sampled or not (Line 11). If the user is classified as an authority, we update the $Top\text{-}K\text{-}Heap$ with the *auth-value* of the user (Line 12), and include the element in our sample (Line 13).

Finally, Rhea features a post-processing step to improve the quality of the sample by filtering-out elements that were wrongly considered as relevant while we were processing the stream (Lines 14-16). This step processes the elements of the sample and removes all those whose respective users are not in the $Top\text{-}K\text{-}Heap$.

[6]Depending on the stream an element may contain more than one positive or negative indicators of importance. We consider this in our implementation but we omit it from the presentation of our algorithm for simplicity.

IV. EXPERIMENTAL EVALUATION

We implemented[7] Rhea using Java. Our evaluation is based on two datasets: i) one that comprises 467 million *tweets* from 20 million users of Twitter (T), covering a period from June 2009 to December 2009 [23], and ii) one that consists of 263,540 answers to 83,423 questions posted by 26,752 users of StackOverflow (SO), between February 18, 2009 and June 7, 2009 [8]. We first present the details of our experimental setting. Then, we proceed with the evaluation of Rhea by answering the following questions:

1) How does Rhea compare against white-list based sampling in terms of *recall*, *precision*, and *F1-score*?
2) Is Rhea able to assess the ranking relevance of the sampled documents?
3) What is the impact of the parameters involved in the execution of Rhea?

A. Experimental Setting

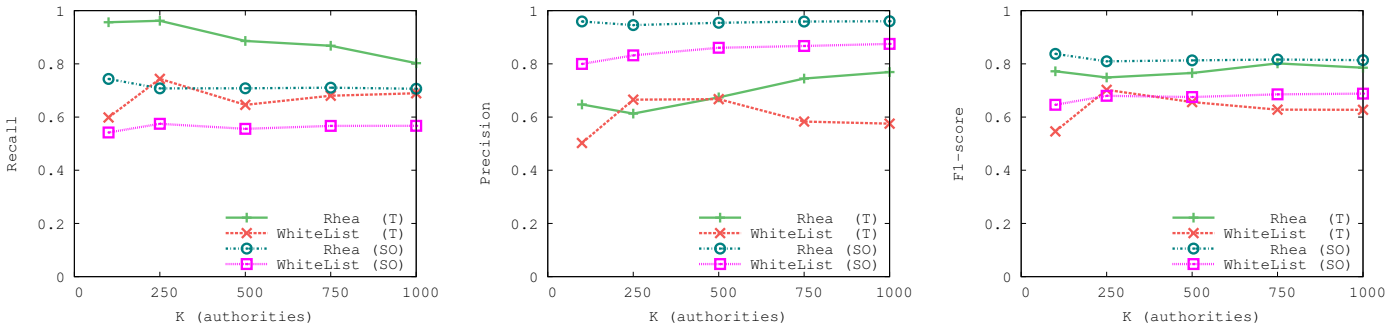
For our Twitter dataset we include in our stream all tweets published during August 2009 – December 2009, i.e., $|S| = 411{,}778{,}304$. Similarly, $|S| = 131{,}768$ for our StackOverflow dataset. We did not use all available elements for the stream, as we also needed a sufficiently large part of the dataset to create static white-lists for a method based in [10] that we compare against. We created the white-lists using the *auth-value* rankings that result from all the available user activity occurring before the activity of the stream. The respective approach decides to sample elements from the stream based on a single criterion: whether the user publishing the element is part of the white-list or not. For the rest of this work we will refer to this method as WhiteList. The elements of both our datasets are timestamped which enables us to replay them chronologically. Unless stated otherwise, Rhea is initialized using the following parameters: i) $p = 0.2$, to use 20% of the stream's elements to extract mentions, and ii) $d = 7, w = 20{,}000$, which gives us 99% confidence that $\epsilon < 0.0001$.

B. Recall, Precision, and F1-score Comparison

We commence our evaluation by comparing the performance of Rhea against WhiteList with regard to *recall*, *precision* and *F1-score* measures. For each element (tweet or answer) we include in our samples, we make a binary assessment concerning the user who posted it. If the user is among the top-K according to the ground-truth, we mark the element as relevant; otherwise, we consider the element to be non-relevant.

We observe in Figure 4(a) that Rhea significantly outperforms WhiteList with regard to *recall* for both datasets. That is, Rhea is able to include *more relevant* documents than WhiteList in its sample. In particular, more than 74% of the relevant documents is included in the sample of Rhea for

[7]Source code and reproducible tests: https://github.com/panagiotisl/rhea



(a) Comparison regarding *recall*. (b) Comparison regarding *precision*. (c) Comparison regarding *F1-score*.

Fig. 4: *Recall*, *precision*, and *F1-score* comparison between our approach and a baseline for our two datasets (`T`, `SO`) when querying for the tweets of the top-100, 250, 500, 750, and 1,000 authorities of the stream.

both datasets even for $K = 1{,}000$, whereas, `WhiteList`'s recall drops as low as 0.55. Figure 4(b) illustrates the *precision* achieved by `Rhea` and `WhiteList`. `Rhea` behaves much better than `WhiteList` for the `StackOverflow` dataset, achieving almost perfect *precision*. For `Twitter`, we observe that both methods initially behave similarly. This is because a few very active non-authorities that are mistakenly taken as authorities may heavily impact *precision* for small values of K. However, as K grows `Rhea` significantly outperforms `WhiteList` for `Twitter` as well. Finally, we illustrate the results of both methods regarding *F1-score*, i.e., the harmonic mean of *precision* and *recall*, in Figure 4(c). We observe that our approach achieves an *F1-score* that is above 0.8 for `StackOverflow` and close to 0.8 for `Twitter` regardless of K. In contrast, using a static white-list, the *F1-score* is much lower and ranges between 0.54 and 0.7.

C. Evaluation of Ranked Retrieval Results

Recall, *precision*, and *F1-score* measures are appropriate for sets of documents that have no ranking information associated to them. The binary assessment we make to classify an element as relevant or non-relevant does not consider the significance of the element with regard to its respective user's authoritativeness. However, we are keenly interested in *ranking quality*. To this end, we employ two additional measures that take under consideration the level of relevance of each element, namely *Spearman's* ρ and *Normalized Discounted Cumulative Gain* (*NDCG*).

1) Evaluation using Spearman's ρ*:* We first investigate the rank correlation between the ground-truth *auth-values* resulting from all the elements of the stream and the *auth-values* derived from each of the two methods examined in this paper. Figure 5 depicts the *Spearman*'s rank correlation for the top-K users of the ground-truth with their respective rankings for `Rhea` and `WhiteList`. In particular, we assigned the rank of 1 to the user with the highest *auth-value* in the ground-truth and increased the rank as we proceeded to users with lower *auth-value*, until the K^{th} user. Then, we created pairs with the rankings of the users that occur when using `Rhea` and `WhiteList`. We observe that there is an *extremely strong correlation* for `Rhea`, i.e., our approach is able to adapt and derive the order of the top-K authorities very accurately. In

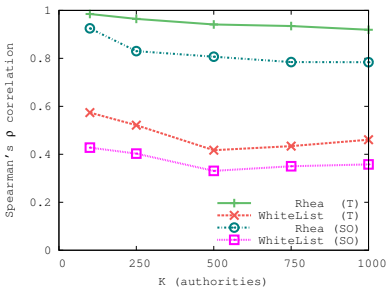


Fig. 5: Comparison of `Rhea` and `WhiteList` on Spearman's ρ for `Twitter` (`T`) and `StackOverflow` (`SO`).

contrast, `WhiteList` exhibits *moderate* to *weak* correlation. These results do exhibit in unambiguous terms the superiority of `Rhea` over contemporary white-list methods, as in addition to higher *precision* and *recall*, our adaptive algorithm captures *much more accurately* the level of importance of each user.

2) Evaluation using NDCG: Next, we use *NDCG*, a measure, suitable for situations of non-binary notions of relevance [14]. *NDCG* is evaluated over some number K of top results. We consider rel_i to be the graded relevance of the result at position i. Then, the *discounted cumulative gain* (*DCG*) at K is defined as:

$$DCG_K = rel_1 + \sum_{i=2}^{K} \frac{rel_i}{log_2(i)} \tag{3}$$

From Eq. (3) we observe that DCG reduces the graded relevance value of each result logarithmically proportional to its respective position in the ranking, to penalize highly relevant documents that appear lower than their actual position [21]. Our goal is not only to retrieve a ranked listed of users according to their authoritativeness, but also retrieve their social activity. Therefore, for our purpose we propose an extension of Eq. (3) that considers the *recall* for each user i:

$$DCG_K = rel_1 * recall_1 + \sum_{i=2}^{K} \frac{rel_i * recall_i}{log_2(i)} \tag{4}$$

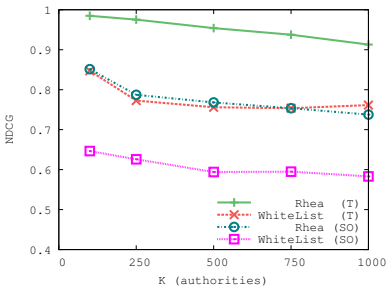


Fig. 6: Comparison of Rhea and WhiteList on NDCG for Twitter (T) and StackOverflow (SO).

NDCG results after normalizing the cumulative gain at each position for a given K as follows:

$$NDCG_K = \frac{DCG_K}{IDCG_K} \tag{5}$$

where $IDCG_K$ is the maximum possible (ideal) DCG for the given set of relevances:

$$IDCG_K = rel_1 + \sum_{i=2}^{|REL|} \frac{rel_i}{log_2(i)} \tag{6}$$

and $|REL|$ stands for the ordered list of relevant documents up to position K.

We consider that the elements of user i have a relevance $rel_i = K + 1 - rank(i)$, where $rank(i)$ is the ranking of the users according to their ground-truth *auth-value*. Thus, the elements of the user with the highest *auth-value* have a relevance of K, whereas those of the user with the K^{th} highest ground-truth ranking have a relevance of 1.

Figure 6 illustrates the results of Rhea and WhiteList with regard to *NDCG* for different values of K. We observe that Rhea again *significantly outperforms* the WhiteList method for both datasets. The latter performs *poorly* with regard to *NDCG* as its value is penalized severely when assigning low rankings to highly relevant users. A vital observation here is the improved performance of Rhea on *NDCG* compared to *recall* (Fig. 7(a)). From this we can induce that the few relevant documents that Rhea is unable to retrieve, are usually not of high relevance. If that was the case, the *NDCG* results would be worse than those measuring *recall*. This is particularly important; we are interested in sampling the elements of the top-K users in the stream, and thus, we are generally more keen on retrieving the elements of the most relevant users. Figure 6 shows that Rhea is *very effective* in doing so.

D. Impact of Techniques and Parameters

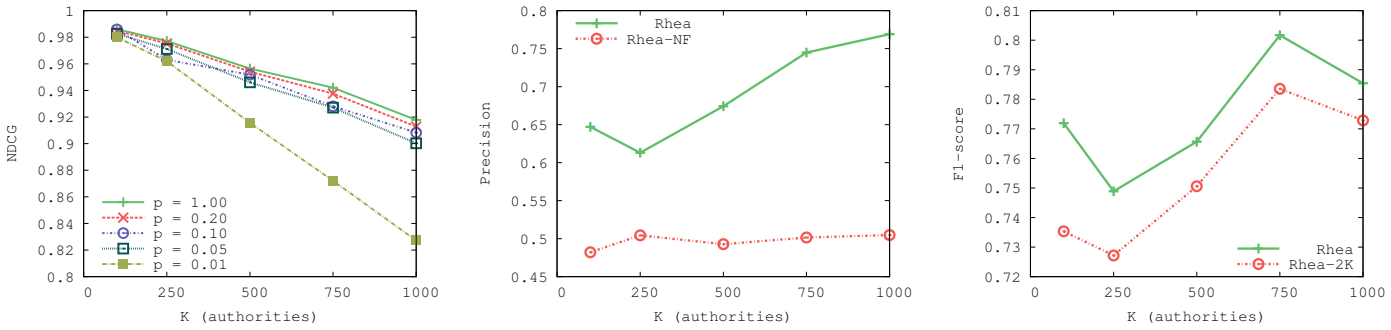
In this section, we investigate the impact of Rhea's techniques and parameters using the largest of our two datasets, namely Twitter. First, we examine the performance of Rhea when altering the probability p of examining a tweet of the stream S to extract mentions and form the *network of authorities*. Second, we quantify the importance of the filtering step of the Rhea algorithm (Lines 14-16 of Algorithm 2). Third, we vary the size of the $Top\text{-}K\text{-}Heap$ to examine its impact on *F1-score*. Our findings are in agreement with those that come up using the StackOverflow dataset, but we omit the latter due to limited space.

1) Varying the Value of Probability p: Rhea involves a random sampling subprocedure, that selects to use with some probability $p \in (0, 1]$ an element of the stream to form the network of authorities. This process significantly reduces the computational overhead of Rhea as we use $|S| * p$ elements of the stream, instead of $|S|$. We examine here the impact this probability has on the results of Rhea with regard to *NDCG*. Figure 7(a) depicts the performance of our algorithm in settings where p is equal to $0.01, 0.05, 0.1, 0.2,$ and 1, respectively. We observe that using a sample of 20% of the stream's elements we are able to achieve performance that is almost as good as that of using the *entire* stream. Moreover, we observe negligible differences when reducing p to 0.1 or 0.05. In fact, the impact of probability p is noticeable only when p is extremely low. Finally, even though using 1% of the elements leads to worse performance, the *NDCG* results we get for Rhea still outperform WhiteList *significantly*. We note that using $p = 0.2$ instead of $p = 1$ greatly reduces processing time. For example, we drop from 3,844 to 2,533 seconds for $K = 100$. For $p = 0.01$ Rhea terminates after 2,189 seconds, slightly over WhiteList that needs 2,040 seconds.

2) Removing the Filtering Step: Rhea samples elements from the stream in a greedy fashion. Therefore, elements of users that are only temporarily part of the top-K authorities manage to end up in our sample. However, when the sampling process is over, we are aware of a final set of top-K authorities, that we have experimentally shown to be a very accurate representation of the actual list of authorities. Hence, we are able to filter-out the elements that in retrospect should not have been collected, by iterating over the sampled elements. We note that $\hat{S} \ll S$, so this operation is inexpensive. Figure 7(b) compares the performance of Rhea when the filtering step is on (Rhea) and off (Rhea-NF). We opt to report the *precision* value, as *recall*, *Spearman's* ρ, and *NDCG* results are all unaffected by this modification. We observe that the difference in *precision* performance is indeed significant. In particular, the difference is *over* 25 percentage points for $K = 1{,}000$, and is never less than 10 percentage points for any K examined.

3) Impact of the Capacity of the Top-K-Heap: We complete our exploration on the parameters of Rhea by examining the impact of the size of the structure we use for holding the stream's current list of authorities. Rhea maintains a heap of authorities induced from the social activity occurring in the stream. This heap has a maximum capacity that enables us to decide on whether to temporarily include an element in our sample or permanently discard it. The size of this heap in our experiments is set to K, i.e., the number of authorities whose activity we aim to include in our sample.

We investigate here an approach that may potentially increase our *recall*. Our intuition is that some authoritative users are *"late bloomers"*, i.e., their importance is not visible until later than expected. Rhea is unable to recover the *entire* activity of such users as it is unaware at the time of sampling of the *final* ranking of each user. However, we may opt to include the activity of more users while sampling, and eventually hold on to the elements produced by those who we believe are the top-K authorities. Figure 7(c) illustrates a comparison of the



(a) Impact of p on NDCG. (b) Impact of the filtering step of Rhea (c) Impact of the $Top\text{-}K\text{-}Heap$ capacity.

Fig. 7: Impact of probability p on *NDCG* (a), of the filtering step of Rhea on *precision* (b), and of the capacity of the $Top\text{-}K\text{-}Heap$ on *F1-score* (c).

performance of Rhea when using a $Top\text{-}K\text{-}Heap$ of capacity K and $2K$, respectively. We use the *F1-score* measure as the capacity of this structure impacts both *recall* and *precision*. We observe that the *F1-score* of Rhea when using a capacity of $2K$ is slightly worse. More specifically, our *recall* is improved as we include the activity of more users during sampling. However, using a larger capacity also leads to including more false positives in our sample. Therefore, we do indeed notice an improvement in *recall*, but it is accompanied with significantly worse *precision*.

V. Related Work

Our work lies in the intersection of social activity stream sampling and authoritative social network users identification. Here, we briefly discuss pertinent efforts in these two areas.

Social Activity Stream Sampling: Related research efforts have mainly focused on the Twitter microblogging service due to its immense popularity and low latency access to its stream of activity. Ghosh et al. [10] compare random samples of Twitter with samples that are taken using a white-list of users. Their motivation is to avoid the large amount of spam, non-topical and conversational tweets that random sampling preserves. The first set of tweets was acquired through the Streaming API, while the second is created using tweets from half million white-listed users. The white-list is derived using Twitter Lists [9], i.e., user generated lists of prominent Twitter accounts. The random sample features a substantially larger population of users, whereas the white-list sample's tweets are extremely more popular. Moreover, the quality of the tweets of the white-listed users is found to be superior. In particular, about 90% of the random sample's *tweets* are conversational, whereas 43% of the white-list sample's tweets contain useful information on a certain topic. Our work is similar to [10] as we also sample streaming social activity content. However, our work does not rely on static white-lists and our focus is not on a specific social network.

Palguna et al. [17] come up with a theoretical formulation for sampling Twitter data. They investigate the number of tweets that is needed to come up with a representative sample using random sampling with replacement. To decide on how representative a sample is, they examine how the frequency of elements in the streams correlates in the sample and the original data. In addition, they examine the case of going through tweets one-by-one and sampling each tweet independent of others with probability p. They show that this behaves similarly to random sampling with replacement and is primarily influenced by the size of the sample. We build on this very last result to speed-up the formation of our network of authorities.

Research efforts have also focused on the quality of the samples offered directly from Twitter. Morstatter et al. [15] perform a comparison of Twitter's Streaming API sample and Twitter's Firehose to examine the impact of the sampling technique of the first. They compare the top *hashtags* of the two datasets, as well as those of random samples taken from the Firehose dataset, and find that the random samples find the top hashtags more consistently than the Streaming API. Moreover, a comparison of topics in the two datasets is performed, using LDA, which shows that decreased coverage in the Streaming API data causes variance in the discovered topics.

Mining streams of social activity is challenging due to the implicit network structure within the stream that ought to be considered along with the content. Aggarwal and Subbian [1] focus on clustering and event detection using social streams. They show that using both the content and the linkage information has numerous advantages. Node counts for individual clusters are handled by employing Count-Min sketches [7]. We also apply Count-Min sketches to summarize counting information of social streams. However, we do not deal with clustering or event detection; rather, we focus on sampling content published by authorities.

Authoritative Users in Online Social Networks: Zhang et al. [25] investigate different network-based ranking algorithms to identify prominent users in a online social network. They show that relative expertise can be automatically determined through structural information of the network, as they find that network-based algorithms perform nearly as good as crowd-sourcing. In addition, they report that simple measures behave at least as good as complex algorithms. In particular, they come up with *z-score*, a measure that considers both the question and answer patterns of a user in a Q & A

community, that best captures the relative expertise of users in a network. In this paper, we rely on the findings of [25] on the effectiveness of *z-score* and propose a generalized version of this measure to identify the top-K authorities in any social activity stream. Agichtein et al. [2] exploit various kinds of community feedback to export high quality content from social media. Among else, they use quality ratings on the content. Pal and Counts [16] use probabilistic clustering and a within-cluster ranking procedure to identify topical authorities on `Twitter`. In an effort to exclude users with high visibility they use nodal features, such as the in-degree. In [5], Bozzon et al. focus on finding topical experts in various popular social networking sites. Their approach takes into account user activity as well as profile information. We operate on a streaming setting and decide whether new content is useful as it becomes available. Therefore, certain aspects of the aforementioned approaches, such as exploiting user ratings, in-degrees, and profile information are not applicable. Ghosh et al. [9] propose `Cognos`, which distinguishes authoritative `Twitter` users using the frequency at which they are included in `Twitter Lists`. This approach assumes the presence of user annotated information indicating importance, whereas in this paper we consider the task of sampling a stream without any prior knowledge. Moreover, we show that static white-list approaches get outdated very quickly and are unable to identify newly emerging authorities. Rybak et al. [19] also point out that authoritativeness is not static. However, they do not deal with stream sampling. Instead, they focus on a co-authorship network and create timestamped profiles of user importance.

VI. Conclusion

In this paper, we propose and implement `Rhea`, the first reported effort to realize adaptive behavior for sampling authoritative content from social activity streams. We commence by exposing the *dynamic* nature of this task which calls for approaches different from employing static white-lists of authoritative users. Then, we proceed by addressing the challenges involved in our dynamic approach. `Rhea` employs Count-Min sketches to compactly maintain both positive and negative indicators of importance of all users appearing in a social activity stream. We additionally propose a novel structure termed $Top\text{-}K\text{-}Heap$, to efficiently query for the top-K authoritative users in the stream, using their relative ranking resulting from their *auth-value*. The latter allows for identifying authoritative users independently of the underlying social network. To reduce the processing overhead of extracting indicators of importance from social activity streams, `Rhea` opts to include in this process each element of the stream with probability p. Finally, `Rhea` features a post-processing step that reevaluates content included in the sample, using the more refined classification model that is available after reading the whole stream.

We compare `Rhea` with a static white-list approach using two datasets reaching up to half a billion posts. We show that `Rhea` exhibits significantly improved performance with regard to both *recall* and *precision*. The superiority of `Rhea` is even more evident when comparing on ranking accuracy, using the Spearman's ρ and *NDCG* measures. Finally, we investigate the effect of various parameters of `Rhea` and ascertain its improved efficiency and effectiveness.

References

[1] C. C. Aggarwal and K. Subbian, "Event detection in social streams," in *SDM 2012*, pp. 624–635.

[2] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," in *WSDM 2008*, pp. 183–194.

[3] M. D. Atkinson, J. Sack, N. Santoro, and T. Strothotte, "Min-max heaps and generalized priority queues," *Commun. ACM*, vol. 29, no. 10, pp. 996–1000, 1986.

[4] M. Bouguessa and L. B. Romdhane, "Identifying authorities in online communities," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 30:1–30:23, 2015.

[5] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci, "Choosing the right crowd: expert finding in social networks," in *EDBT '13*, pp. 637–648.

[6] A. Ching, S. Edunov, M. Kabiljo, D. Logothetis, and S. Muthukrishnan, "One trillion edges: Graph processing at facebook-scale," *PVLDB*, vol. 8, no. 12, pp. 1804–1815, 2015.

[7] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *J. Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[8] C. DuBois. (2009, Jun.) Stackoverflow data. [Online]. Available: https://www.ics.uci.edu/~duboisc/stackoverflow/

[9] S. Ghosh, N. K. Sharma, F. Benevenuto, N. Ganguly, and P. K. Gummadi, "Cognos: crowdsourcing search for topic experts in microblogs," in *SIGIR '12*, pp. 575–590.

[10] S. Ghosh, M. B. Zafar, P. Bhattacharya, N. K. Sharma, N. Ganguly, and P. K. Gummadi, "On sampling the wisdom of crowds: random vs. expert sampling of the twitter stream," in *CIKM'13*, pp. 1739–1744.

[11] P. Jurczyk and E. Agichtein, "Discovering authorities in question answer communities by using link analysis," in *CIKM 2007*, pp. 919–922.

[12] J. M. Kleinberg, "Hubs, authorities, and communities," *ACM Comput. Surv.*, vol. 31, no. 4es, p. 5, 1999.

[13] P. Lawrence, B. Sergey, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford University, Technical Report, 1998.

[14] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[15] 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 2013*.

[16] A. Pal and S. Counts, "Identifying topical authorities in microblogs," in *WSDM 2011*, pp. 45–54.

[17] D. S. Palguna, V. Joshi, V. T. Chakaravarthy, R. Kothari, and L. V. Subramaniam, "Analysis of sampling algorithms for twitter," in *IJCAI 2015*, pp. 967–973.

[18] O. Papapetrou, M. N. Garofalakis, and A. Deligiannakis, "Sketch-based querying of distributed sliding-window data streams," *PVLDB*, vol. 5, no. 10, pp. 992–1003, 2012.

[19] J. Rybak, K. Balog, and K. Nørvåg, "Expertime: tracking expertise over time," in *SIGIR '14*, pp. 1273–1274.

[20] C. Wagner, V. Liao, P. Pirolli, L. Nelson, and M. Strohmaier, "It's not in their tweets: Modeling topical expertise of twitter users," in *PASSAT 2012, and SocialCom 2012*, pp. 91–100.

[21] Y. Wang, L. Wang, Y. Li, D. He, and T. Liu, "A theoretical analysis of NDCG type ranking measures," in *COLT 2013*, pp. 25–54.

[22] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge University Press, 1994, vol. 8.

[23] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *WSDM 2011*, pp. 177–186.

[24] M. B. Zafar, P. Bhattacharya, N. Ganguly, S. Ghosh, and K. P. Gummadi, "On the wisdom of experts vs. crowds: Discovering trustworthy topical news in microblogs," in *CSCW 2016*, pp. 437–450.

[25] J. Zhang, M. S. Ackerman, and L. A. Adamic, "Expertise networks in online communities: structure and algorithms," in *WWW 2007*, pp. 221–230.