Context Based Algorithm for Social Influence Measurement on Twitter

Alaa Alsaig(✉), Ammar Alsaig, Marwah Alsadun, and Soudabeh Barghi

Computer Science and Software Engineering, Concordia University, Montreal, Canada
{al_alsai,a_alsaig,m_alsadu,s_arghi}@encs.concordia.ca

Abstract. The social media became one of the most effective method for marketing and for information propagation. Therefore, measuring users influence is important for organizations to know which user to target to successfully spread a piece of information. Twitter is one of the social media tools that is used for information propagation. The current methods for measuring influence of Twitters users, use ranking algorithms that focus on specific criteria such as number of followers or tweets. However, different cases creates different needs in measuring influence. Each need could include different elements with different priority. One of these cases is local businesses which need to propagate information within a specific context such as location. That is, the most influential user for such a business is the one that has the highest number of followers that are located within the required location. Therefore, in this paper, we use the X algorithm for measuring users influence on Twitter by ranking users based on followers context that is represented by number of elements. Each element is given a weight to prioritize elements based on client demand.

Keywords: Social influence measurement · Context
Twitter users influence

1 Introduction

Information propagation is an important process for many tasks of many sectors such as marketing, awareness, and news propagation. In order to ease the task of information propagation, social influence measurement becomes useful to provide us with the most influential nodes to be targeted by those who needs this information. To measure influence in social networks, users influential could be defined, and hence measured, differently considering different aspect. An Influential user can affect other users' actions [1], or change behavior, cause effect in online social networks [2]. Others focus on influential users in term of spread information. They see influential users as the most users who can spread information in the social network [1] and [3]. Furthermore, some went to classify influential users into many classification based on the context of influence, for

P. Cong Vinh and V. Alagar (Eds.): ICCASA 2018/ICTCC 2018, LNICST 266, pp. 136–149, 2019.
https://doi.org/10.1007/978-3-030-06152-4_12

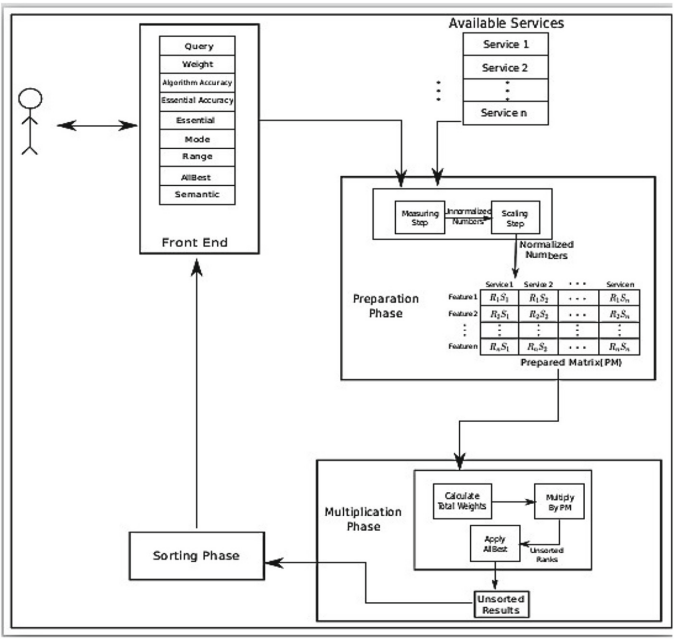

Fig. 1. XAlgorithm

example, opinion leaders, inventors (start new topics), celebrities, spreaders, disseminators, engager, connectors, etc [1]. This provides no agreed definition for influence in social networks.

Twitter is a social media tool that is used to propagate information through its users. Many work has been done to measure influence of Twitter users [1]. From this study, the social influence of Twitter's users is measured using different algorithms that consider all or some of the following elements:

- the number of followers
- number of tweets
- number of retweets
- Social influence is measured

In fact, the previous review gives intuitive idea that influence can be perceived from many different contexts.

2 Problem Definition

For twitter, usually deal with metrics that consider retweets, mentions, and number of followers [1]. However, the criteria for measuring influential users are as many as the growing number of techniques that rank influential users.

In fact, some other contextual factors play important role in measuring influence as location, vulnerability etc. [4].

To the best of our knowledge, there is no one algorithm that can measures influence in Twitter based on many different criteria. In our project, we consider information propagation within a specific area which requires contexts as criteria for influence measurement. Therefore, it is needed to find an algorithm that is able to suit different needs/criteria of measuring influence.

3 Paper Structure

This paper, starts with the literature review in Sect. 4. Contribution is explained in V. VI includes the methods, the implementation, and evaluation of our method. Limitation and conclusion provided in Sects. 7 and 8 consecutively.

4 Literature Review

Through the papers, we want it to find an algorithm that has the features that matches our need. Below, the founded algorithms are described below:

- textbfUserRank [5] User influence is measured in such a way that the more high ranked followers a user has, the higher user rank for this user. Hence the focus is more on the number of followers, however it has been shown that small number of followers is worth more than a large number of followers if the former are more active users [3]. Therefore, we can see that there is another metrics, which is activeness, could have been considered to give more accurate result. In fact, that emphasizes the need of consider many criteria when measure influential users in social networks
- textbfTrueTop [2] The algorithm of TrueTop, basically filters the list of users that are ranked as influential users to identify real users (non Sybil users) based on the incoming retweets, replies, and mentions each Twitter user has. Again this algorithm doesn't give the flexibility we are looking for because it has limited metrics to measure influence and doesn't prioritize them.
- textbfContent and Conversion [6] This method uses content and conversation metrics Based on number of tweets, mentions, retweets and replies and two Although it prioritize conversation and content by giving conversation more weight, considering it as more influential, but they only consider two concepts content and conversation.
- textbfThe X Algorithm [7] The X algorithm is a usercentric algorithm for ranking services Fig. 1. That is, the algorithm ranks services in the system based on the similarity of each service to the user query (requirement). The more similar the service is the higher rank it will get. The algorithm has a lot of features that motivated us to use it for our project, below are the most important features of the X algorithm:
  (1) Supporting different data type: Boolean, string, and numerical values.
  (2) The results manipulated with different modes such as Best/Exact Modes.

(3) Elements interpreted into different semantics: More is Better, Less is Better and Exact is Better semantics.
(4) Consistent outputs because it is based on provided requirements.
(5) X-Algorithm is not restricted to specific number of elements.
(6) Provide high performance.

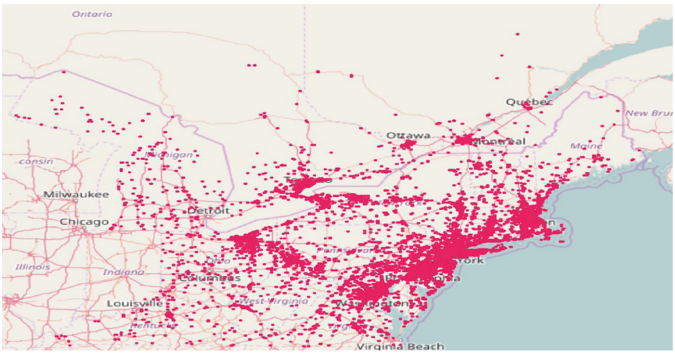

Fig. 2. Area of gathered data

```
type Tweet struct {
    TweetId           int64      `json:"tweet_id"`
    TweetFavoriteCount int       `json:"tweet_favorite_count"`
    TweetRetweetCount  int       `json:"tweet_retweet_count"`
    TweetCoordinates   *geo.Point `json:"tweet_coordinates"`
    UserId            int64      `json:"user_id"`
}
```

Fig. 3. Data structure corresponding to a tweet

5 Contribution

In this project, we provide a new method for measuring influence with consideration to any number of elements with high efficiency. The method includes the ability to give a weight for each element depending on our priority.

Particularly, for our case we consider user's context that includes location, number of followers, and number of friends, as criteria for social influence measurement.

6 Proposed Solution

The solution goes through different phases that are described below:

6.1 Phase 1: Data Gathering

Gathering data has gone through six different steps described as follows:

```
type User struct {
    FollowersCount int    `json:"followers_count"`
    FriendsCount   int    `json:"friends_count"`
    ID             int64  `json:"id"`
    Location       string `json:"location"`
    StatusesCount  int    `json:"statuses_count"`
}

type UserFollowers struct {
    User        *User   `json:"user"`
    FollowerIDs []int64 `json:"follower_ids"`
}
```

Fig. 4. Data structure corresponding to a user and its followers

(1) **Fetching tweets from the Twitter Stream API:** During two days of time, we fetched tweets from the Twitter Stream API, using a location filter parameter that restricted the area as in Fig. 2. The South West coordinates was set to 37.916, 82.182, while the North East coordinate is 49.774 69.700. During those two days 16,714 unique users tweeted a total of 41,403 tweets. All the data was gathered in JSON Lines files. JSON Lines format [8]. The JSON Lines format is essentially JSON Objects separated by new lines. It proved to be very beneficial, to work with the stream data, and to be able to do further operations on the files using basic Unix tools such as wc, split, tail, cut or head [9].
It is worth to note that we used a Golang library which is called go-twitter[10], to do all the operations with the Twitter API, the library offers a convenient client to interact with the Twitter API, and offers data structure to represents the different entities in Twitter Fig. 3.

(2) **Gathering User's Followes IDs:** Using the 16,714 followers, we go in the previous step, we gathered their followers IDs using the Twitter REST API. This was a very expensive operation. Twitter's API limits those call at 60 requests per hour, and it gives us only a maximum of 5000 IDs at a time. Therefore we decided to only make our analysis on users who have 5000 followers or less. This is of course, a great limitation, but can only alleviated with time.
However, we used Golang's builtin concurrency system to parallelize the API calls among 4 different accounts. At the end of this operation, we had 9,254,314 followers IDs, in a 4.3 GB file.

(3) **Accumulating user's information:** We needed the information of the followers to do our analysis, so we accumulated their own information using their ID that we gathered in the previous step Fig. 4. For this to be done, we made some batch HTTP request, Twitter is much more generous with this operation (1200 user per hour). It is worth noticing, that Twitter gives us much more information about the User, but for a storage purpose, we only kept the data with the most importance to this project [8,10].

(4) **Using Geocoding APIs to geocode User's location:** While Tweets have exact coordinates, as shown in Fig. 3, users only have a location represented as a string. This created another challenge: we needed to geocode the 9,254,314 Users [11,12]. For this purpose, we have used several APIs described in Table 1:
Once we had a sizable amount of geocoded location, we realized that some of the users had very similar locations, for example: "New York" "New York." "New York, NY "New York, NY" "New York NY!?" "NYC" As we can see, locations only differ by punctuations, or emojis. We therefore decided, to employ a Fuzzy Location Match, using Levenshtein distance. The program written in Golan ran on a Kubernetes Cluster on Google Cloud Platform with three nodes of type n1standard4. The program had to compare 6,500,000 UTF8 strings, with a set of 247,049 know location [13,16]. It ran for an entire day, at 100% CPU usage, and resulted in 490,340 new locations. To this date, we have a database of 1,190,068 unique locations, which covers 7,333,088 users.

(5) **Refetching the Tweets:**
One month after the tweets were tweeted, we refetched the tweets by ID, using another batch HTTP request (1200 tweets/hour). From there, we could see the number of retweets and the number of favorites the tweets got.
However, the numbers of retweets and favorites were really small. Over 14027 tweets, only 1297 tweets were retweeted and 5627 has a favorite count. The reason those numbers are really small is probably due to the very loose restrictions and filters on the tweets that we streamed in the first place. An improvement would been to filter tweets based on a minimum of retweets and favorites, then do the full analysis.

(6) **Calculate Distance:**
The remaining task was to compute the distance between the users and get the data ready for the XAlgorithm. The distance between two coordinates was computed using the Haversine distance. It is essentially, the great circle distance between two points on a sphere given their longitudes and latitudes. We also created an HTTP Client to make the request to the XAlgorithm HTTP Server. The only resource we had was a POST on /query with the following JSON body Fig. 5.

Table 1. Used geocoding APIs

API	Number of account(s)	Limits
Google geocoding [13]	1	2,500 requests/day
Location IQ [14]	2	10,000 requests/day
MapBox [11]	1	5,000 requests/day
OpenCage [12]	1	2,500 requests/day
MapQuest [15]	3	15,000 requests/day

6.2 Phase 2: Implementation

Before we go into the details of the implementation, it is important to understand how the X algorithm works Fig. 6. The X algorithm takes a query as an input that includes the required value and the weight (priority) for each field. Then, the algorithm based on the similarity between the items it has in the system to the given requirement will rank the items. The more similar, the higher rank is assigned. The X algorithm has two modes, one is the best mode and the other is the exact mode. The best mode provides the better items than the entered specifications. This mode is selected in our project because it does suit our requirement. The entered specification is the minimum specification that describes an influential for a local business case. Yet, time limit was a barrier for thorough study for logical and accurate justification. The attributes could be inter-operated in different semantics. There are two semantics considered in this project which are explained below:

```
{
    "prop_n": ["followers", "friends", "distance"],
    "rlp": ["followers"],
    "consumer_query": [5000, 100, 1],
    "consumer_rate": [0.005, 0.001, 0.004],
    "input": [[20.0, 5.0, 2.0], ......]
}
```

Fig. 5. JSON query to XAlgorithm server

- More is better (MB): is the attribute that is quality like. For example, if the required number of followers is 5000, then a user with 10000 followers is definitely better.

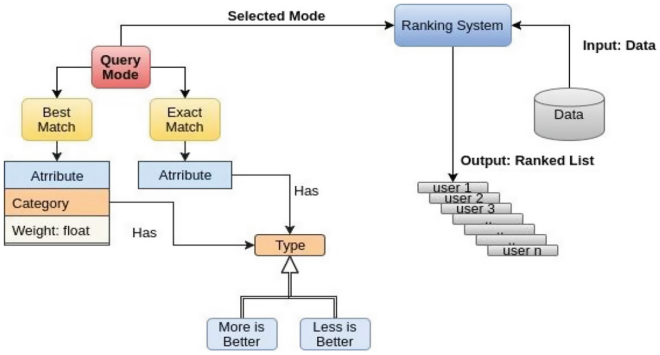

Fig. 6. Abstract XAlgorithm model

- Less is better (LB): is the attribute that is cost like. For example, if the maximum distance between a user and his/her follower is 10 Km then a user with 5 Km distance is better.

Each attribute is prioritized differently with weight element. The weight ranges from 1 to 5. 5 is the highest. The weight value depends on the priority you give for an attribute over another.

In this project, we used five attributes with different semantics and weight as explained in Tables 2 and 3. Hence, to measure influence of Twitter users within a specific area, the implemented program goes through two rounds to finish measurement Fig. 7. The explanation of both rounds are provided below:

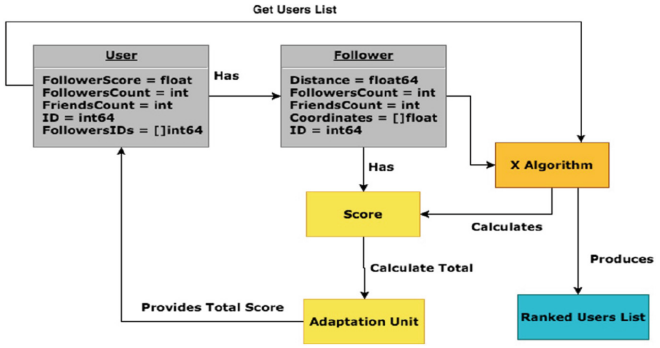

Fig. 7. UML context based algorithm for ranking influence measurement

- **Round 1:** the goal of this round is to provide a ranking score for each follower of all users of Twitter. The ranking score is based on three criteria, which are followers count, friends count, and distance. The semantics of each element provided in Table 2. The requirement (query) we provided for this round is: (number of followers maxFollowers, number of friends 100, and distance 1 km).

The algorithm gives higher score for equal or better of our query and lower score for followers that do not meet the query specification. The first round is applied on all followers of all users which is in total 9,000,000 followers. When the round ends, each follower is assigned a score by the X algorithm that is based on follower similarity to the provided specification. The scores are ranged between [0.54:0.54]. The followers' scores of each user go to the adaptation unit for manipulation. The manipulation process gets rid of all negative scores by adding to each score the value 0.54, which makes scores are ranged between [0:1.08]. Then, the adaptation unit sums all the scores of followers to assign each user with the total score of his/her followers.

- **Round 2:** the goal of this round is to rank the users of Twitter themselves. The ranking is based on two criteria which are the total score (the result of round 1) and the number of friends. The semantics of each of the elements provided in Table 3. The requirement (query) we provided for this round is: (FollowerScores = max(followerScore), and number of friends 100). The result of round 2 is a list that includes all users in our data ranked in order. We have 14000 user and their results illustrated in Fig. 8.

6.3 Phase3: Evaluation

There are three methods of evaluation that are applied, which are explained below:

Table 2. First round attributes (semantics and weight)

Attributes	Weight
FollowersCount (MB)	5
Friends (LB)	2
Distance (LB)	1

Table 3. Second round attributes (semantics and weight)

Attribute	Weight
FollowerScores (MB)	5
Friends (LB)	2

Table 4. Sample 1 for data evaluation

	User 1	User 2	User 3
Follower 1	0.546000	0.552000	0.546976
Follower 2	0.542933	0.549333	0.546329
Follower 3	0.540266	0.546666	0.546044
Rank result	Third	First	Second

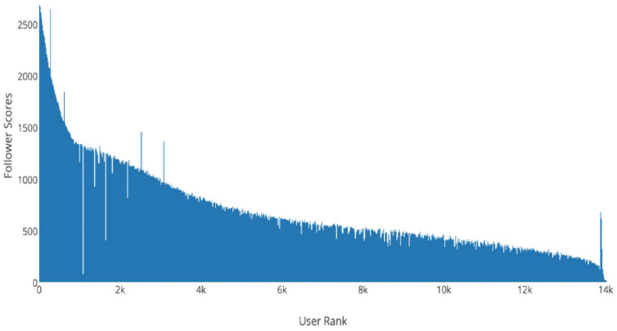

Fig. 8. 14000 ranked users

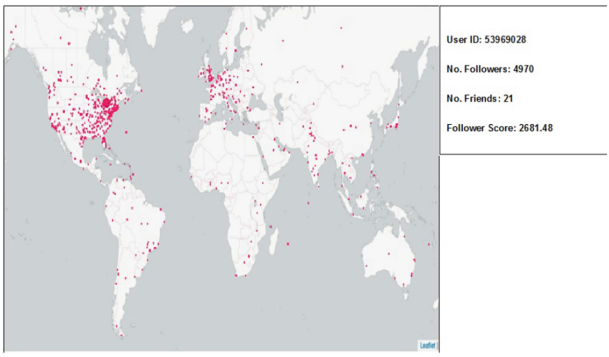

Fig. 9. First ranked user

- **Small Samples:** The algorithm is applied on predefined data to make sure it provides the expected results. In Table 4, we provided user 2 with the very good followers as they have high number of followers, low number of followers, and short distance according to the user's location. User 2 has high number

Table 5. Sample2 for data evaluation

	User 1	User 2	User 3
Follower 1	0.546000	0.552000	0.546976
Follower 2	0.544494	0.549333	0.546329
Follower 3	0.544444	0.546976	0.546044
Follower 4	0.544255	0.546666	0.539110
Follower 5	0.542933	0.546329	0.537855
Follower 6	0.540266	0.546044	0.537636
Rank result	Second	First	Third

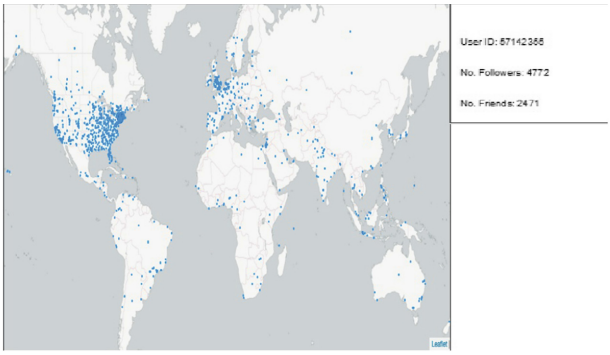

Fig. 10. Fiftieth ranked user

Fig. 11. Hundredth ranked user

of followers but all in far places and user 3 has lower number of followers compare to Users 1 and 2, yet, they live in close area to their user. The results showed what we have expected, as User 2 is considered more influential to our requirement than User1. In Table 5, we still keep User 2 as the optimal user. However, user 1 had the same number of good followers as User 3 but also has bad followers. User three has most of his followers good in terms of distance but not in terms of the number of followers. As we weight the number of followers more than the distance and as User 1 has the same number of good users as User 3, then the algorithm met our expectation by providing User 1 a higher rank than User 3.

- **Ranked User Illustration on Map:** In the following graph shows the geographical distribution of three different users ranked by aforementioned method. Each user is spotted on the map with his/her followers. In Fig. 9, the first ranked user is showing the most of his/her followers are surrounding him/her as most of them live close to the user which is matching our requirement. However, in Fig. 10, the 50th user is punished to be ranked in this position due to having high number of friends.

In Fig. 11, the user is ranked 100th because the specification of this user specially in terms of number of friends, followers and the total follower score do not match with the specification of this study, as a result this user gets a lower rank in general.
In Fig. 12, the three previous users are represented in one map. This is to show the difference User ranked 1, User ranked 50, and User ranked 100. It is obvious, the 50th user (blue color) his friends are scattered and not surrounding in his/her area. The 100th user (Green color) has fewer followers around himself/herself compared to the first ranked user (Red color)

- **Retweet and Favorites:** Keeping track of gathered tweets after one month shows 17% of retweets numbers just related to the first 1000 ranked user though this study method Fig. 13. To make it clearer, this population contains 7% of the whole gathered data user which means these ranked users' tweets has potential influence in term of being retweeted by their followers. User's tweet after one month, about 14000 users. Number of favorites of tweets released by the first 1000 ranked users consist of 16%. The density of the graphs for the first 1000 ranked users shows their potential influence Fig. 14.

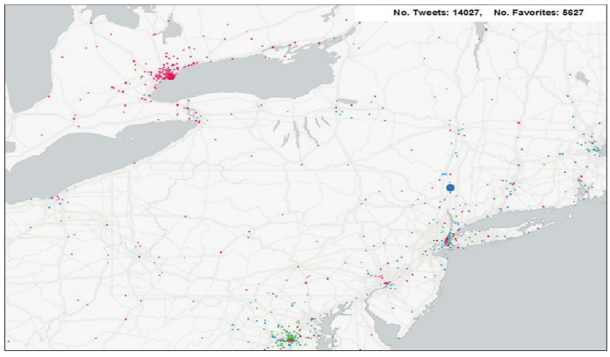

Fig. 12. Map includes the three ranked users (1, 50, and 100) (Color figure online)

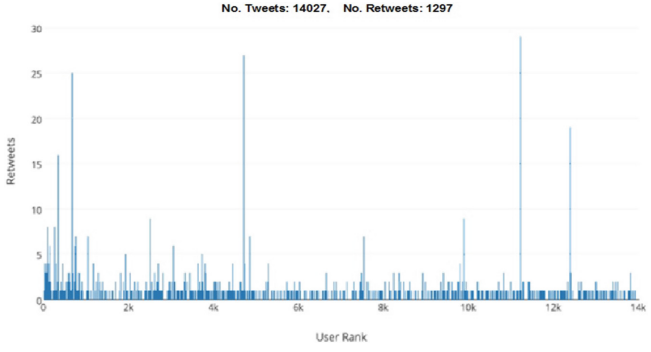

Fig. 13. Retweets of all users

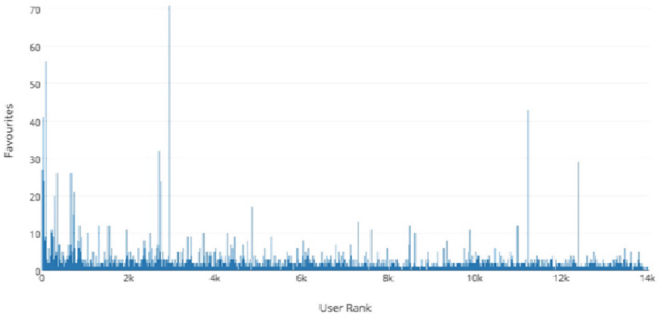

Fig. 14. Favorites of all users

7 Challenges and Limitations

Considering Twitter Laws and rules, our evaluation is limited to only users with at most 5000 followers since the accessibility to more than this is prevented. In addition, the recursive nature of data in Twitter to just two levels depth. The result of this study can only justify the defined scale in this project. Therefore, having different outcome when applying X-Algorithm on the huge real scale of Twitter is expected. Another obstacle we faced in this project is finding location information for all users, which is one of the context elements in our dataset. In the retrieved data, location information of some users is unavailable. To make the effect of this barrier less, we assigned a lower weight to this attribute. Having the mentioned limitations did not help to propagate a specific tweet in reality in order to do more precise evaluation for the proposed method.

However, this study proves that having huge number of followers does not guarantee that the user is influential. The context of a user and their followers is essential to specify the potential influential user. The proposed method provides the flexibility to define context with any element (with different type) with the ability to prioritize these elements using weight. Therefore, it is believed that this method can perfectly find those influential users that can be a profitable targets who serve business purposes. This is useful for all businesses especially local and small ones.

8 Conclusion

Different needs create different ways in measuring influence. Focusing on one context can provide some results which are not fulfilling the main demand. Using a flexible algorithm to consider different criteria not only can overcome this problem, but also it can present much more suitable result according to the need. Not every criterion is as important as the other one however the collaboration of them could result in different rank of user's influence. Therefore, in this paper, we provided a method that uses the XAlgorithm to rank users of Twitter based on the context of user's followers. The ranking results provided promising results

as the retweet and the favorite evaluation method showed reasonable number of retweet and favorite for the first 1000 ranked users. With consideration to the dataset limitation we had, we believe that our method could be an effective method for influence measurement for different needs.

As a future work, many cases or needs could use our method for measuring influence not only on Twitter but any other social media tools. The method we provided is flexible enough to adapt to the needs of any organizations or businesses.

References

1. Riquelme, F., González-Cantergiani, P.: Measuring user influence on twitter: a survey. Inf. Process. Manage. **52**(5), 949–975 (2016)
2. Zhang, J., Zhang, R., Sun, J., Zhang, Y., Zhang, C.: Truetop: a sybil-resilient system for user influence measurement on twitter. IEEE/ACM Trans. Netw. **24**(5), 2834–2846 (2016)
3. Anger, I., Kittl, C.: Measuring influence on twitter. In: Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies, p. 31. ACM (2011)
4. Li, Y., Ding, Z., Zhang, X., Liu, B., Zhang, W.: Confirmatory analysis on influencing factors when mention users in twitter. In: Morishima, A., Chang, L., Fu, T.Z.J., Liu, K., Yang, X., Zhu, J., Zhang, R., Zhang, W., Zhang, Z. (eds.) APWeb 2016. LNCS, vol. 9865, pp. 112–121. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45835-9_10
5. Majer, T., Šimko, M.: Leveraging microblogs for resource ranking. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 518–529. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27660-6_42
6. Hatcher, D., Bawa, G.S., de Ville, B.: How you can identify influencers in SAS® social media analysis (and why it matters). In: SAS Global Forum, pp. 4–7. Citeseer (2011)
7. Alsaig, A., Alagar, V., Mohammad, M., Alhalabi, W.: A user-centric semantic-based algorithm for ranking services: design and analysis. SOCA **11**(1), 101–120 (2017)
8. T.D. Team, dghubble/go-twitter (2017). https://github.com/dghubble/go-twitter
9. Jsonlines.org, Json lines (2017). http://jsonlines.org/
10. T.D. Team, Twitter api overview (2017). https://dev.twitter.com/overview/api
11. M.D. Team, Geocoding, mapbox (2017). https://www.mapbox.com/geocoding/
12. O. Geocoder, Easy, open, worldwide, affordable geocoding (2017). https://geocoder.opencagedata.com/
13. G. Developers, Developer's guide, Google maps geocoding api, Google developers (2017). https://developers.google.com/maps/documentation/geocoding/intro
14. Locationiq.org, Locationiq free and fast geocoding and reverse geocoding service from unwired labs (2017). https://locationiq.org/
15. M.A. Documentation, Geocoding api overview (2017). https://geocoder.opencagedata.com/
16. G.C. Platform, Machine types, compute engine documentation, Google cloud platform (2017). https://cloud.google.com/compute/docs/machine-types