Buzzer Detection on Twitter Using Modified Eigenvector Centrality

Mario Tressa Juzar
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
mariotj.tj@gmail.com

Saiful Akbar
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
saiful@informatika.org

***Abstract*—Social media is an online media where its users can easily participate, share, and create contents. One of the most used social media is twitter. Twitter nowadays used by billions of people to interact with other people. One of the phenomenon that we can observe in social media is user that has influence to other users, which commonly called influencer or buzzer. Buzzer often considered as central point of information spreading, which mean we can analyze it by using centrality analysis. Buzzer detection is one of problem that happen in social media that can be approach by using centrality analysis. One of the centrality analysis method is eigenvector centrality. Dynamics data that occur on twitter can be used as weight in eigenvector centrality and we made some modification in eigenvector centrality. On this paper, we propose a method by using modified eigenvector centrality to detect buzzer by considering dynamics data that occur on twitter.**

Keywords—social media, twitter, buzzer detection, centrality analysis, eigenvector centrality, modified eigenvector centrality

I. Introduction

Social media are interactive computer-mediated technologies that facilitate the creation and sharing of information, ideas, career interests and other forms of expression via virtual communities and networks [1]. In social media, users can easily participate, share, and create content. Based on statistic, users of social media around 2.3 billion people in the world [2].

A big data from social media reses challenges to analyze it. And this area known as social media analytics. Social media analytics is the practice of gathering data from social media websites and analyzing that data using social media analytics tools to make business decisions. The most common use of social media analytics is to mine customer sentiment to support marketing and customer service activities [3].

In social media, there is phenomenon that we can observe which called buzzer. Buzzer is social media user that has influence to other users. Buzzer also considered as central point of friendship in social media. As a central point, we can approach it by using centrality analysis to find the recommended buzzer.

Centrality analysis have 4 main calculation method, which are degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality. These standard calculations does not include dynamics that occur in social media. Although other advanced research make modification by using weight, but the weight still not from the dynamics from social media.

From previous research, posited that utilizing edge weights in the problem of global social search could yield more effective results than a search based on binary ties alone [4]. On this paper, we use the dynamics data from social media and include it in buzzer calculation.

Data from social media is used to create a data model to be used in this research. The standard method eigenvector centrality then modified to use dynamics data from social media as weight. The modified eigenvector centrality then will be the basis for calculate buzzer value for each user node.

II. Related Work

A. Eigenvector Centrality

In graph theory, eigenvector centrality (also called eigencentrality) is a measure of the influence of a node in a network. Relative scores are assigned to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores.

Eigenvector centrality is used to calculate the important of a node on a graph. The node connected to the important node will be considered as a more important node. For example a small twitter account with a few followers but followed by an account that has a lot of followers, if people who have many followers to follow the account that means there is a tendency that account is considered important. The eigenvector centrality calculation is written with the following equation:

$$C_E(v_i) = \frac{1}{\lambda}\sum_j A_{ij}C_E(v_j) \tag{1}$$

$$Ax = \lambda x \tag{2}$$

978-1-5386-7259-4/18/$31.00 ©2018 IEEE

Where:

- C_E = *eigenvector centrality*
- v_i = vertex-i
- $\lambda =$ *eigenvalue*
- A = adjacency matrix
- x = matrix of eigenvector centrality

III. Data Model

Social media data that we used is social media for friendship like facebook, instagram, or twitter. Because of its limitation we don't use facebook data and instagram. Because of that, on this research we use data from twitter.

Data is one of the important things to detect buzzer in social media. First, we analyze what data that should we have to calculate and detect buzzer from social media. On this research, we use data from twitter and got it from twitter API. From twitter data, then we created a data model that will be used in this research. The data model is shown by Fig 1.

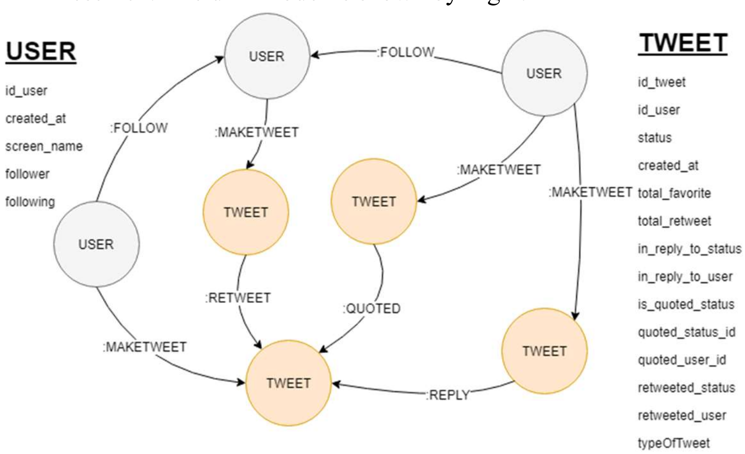

Fig 1 Data model

From this data model, we can observe dynamics that occur on twitter like interaction within user by tweet, follow relation, retweet tweet by user, a user that reply other user tweet, and a quoted tweet that user made. This data model is enough to represent dynamics on twitter. Then, this data model will be used in this reasearh for calculating modified eigenvector centrality and buzzer value.

IV. Buzzer Value With Modified Eigenvector Centrality

The basic concept that we used is eigenvector centrality. Eigenvector centrality is a measure of the influence of a node in network. Relative scores are assigned to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores.

From eigenvector centrality method, we make modification in adjacency matrix so that the contents of the matrix are the weight of the social media graph. And we remove the eigenvalue calculation so that it make the calculation of modified eigenvector simpler. In this weight calculation, we also make another calculation like influence value, activeness value, and node value based on dynamics that occur on twitter.

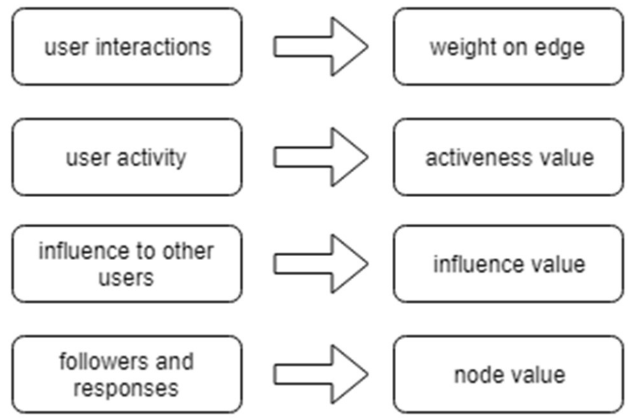

Fig 2 dynamics data mapping

From Fig *2*, we can see the mapping from the dynamics that into the calculation that have been made. This calculation will be used in modified eigenvector centrality, and then detect the buzzer by calculating the buzzer value.

A. Weight Calculation

Friendship on social media between people with one another has different values. People who are closely related often interact on social media, in this case twitter, people can respond by retweeting, replying, or making quoted tweets on tweets made by others.

Based on the interaction and available data, we then made weight calculations on social media graphs that involved relationships between users, through interactions on tweets made by users.

Weight values on social media graphs are taken from the value of the relationship between users on Twitter obtained through interactions between users, namely retweet, quoted tweets, and reply. The '1' value is considered as a relation value, which mean if user followed someone it has the relation value, on the other hand if user does not follow someone the value will be 0. The weight values on this graph are then formulated as follows.

$$w_{(A\rightarrow B)} = 1 + in_{(A\rightarrow B)} \tag{3}$$

$$in_{(A\rightarrow B)} = \alpha \sum r_{(A\rightarrow B)} + \beta \sum q_{(A\rightarrow B)} + \gamma \sum p_{(A\rightarrow B)} \tag{4}$$

$$aw_{(A\rightarrow B)} = \frac{w_{(A\rightarrow B)}}{H(w)} \tag{5}$$

Where:

- $w_{(A \to B)}$ = weight value, which mean A follow B
- $in_{(A \to B)}$ = interaction that happened from A to B
- $r_{(A \to B)}$ = *retweet* from A to B's tweet
- $q_{(A \to B)}$ = *quoted status* from A to B's tweet
- $p_{(A \to B)}$ = *reply* from A to B's tweet
- α, β, γ = portion parameter for calculation, where $\alpha + \beta + \gamma = 1$ and $\alpha > 0,\ \beta > 0,\ \gamma > 0$
- $aw_{(A \to B)}$ = *adjusted weight* from A to B
- $H(w)$ = highest value among all weight

B. Activeness Value

Users in social media have different levels of activeness on social media. There are users who are very active interacting on social media and some are just getting information from the timeline without any desire to be active in the spreading of information or interact. This proves that every social media user has different activeness values.

Activeness value seen from tweet that they have made on particular time. Activeness value is calculated by using time frame, which mean for particular time from first observe until las observe time. Activeness value attached in each user. This value then formulated as follows.

$$av_{(i)} = \frac{\sum_{j=\,start\ date}^{end\ date} t_{(i,j)}}{tf} \tag{6}$$

Where:

- $av_{(i)}$ = *activeness value* on user node *i*
- $t_{(i,j)}$ = valued 1 if user *i* made *tweet* on date *j*, valued 0 if user *i* not made *tweet* on date *j*
- $tf = time\ frame = \ end\ date - start\ date + 1$, in day unit

C. Influence Value

In the dissemination/spreading of information on social media by buzzer, the important thing is how influential users are to other users. The power of this influence will affect other users to spread information. The greater the effect, the greater the number of other users will respond/react to information spread by a user.

Influence has 2 values, namely the value that attached to each user and the value that attached in relation among user. Calculation of the influence value on each user on social media is formulated as follows. Equation (5) only applies for $H\left(in_{(A)}\right) > 0$. When $H\left(in_{(A)}\right) = 0$, the value $ivu_{(A \to B)} = 0$.

$$ivu_{(A \to B)} = \frac{in_{(B \to A)}}{H(in_{(A)})} \tag{7}$$

$$iv_{(A)} = \frac{\sum_{j}^{n} ivu_{(A \to j)}}{n} \tag{8}$$

Where:

- $ivu_{(A \to B)}$ = *influence value user*, from A to B
- $in_{(B \to A)}$ = weight value from edge B which follow A
- $H(in_{(A)})$ = highest weight value among all weight value that following A
- $iv_{(A)}$ = *influence value* on A
- n = total followers of A

D. Node Value

A node value is the value on a node of a friendship graph that represents the condition of the node. Node values vary in each node because they are affected by the relationship with the surrounding node.

To see how the condition of a user node is on a friendship graph on social media, what to be considered is how the user is positioned in the graph and how the surrounding followers respond by retweeting and favorites over the original tweets. Users who have a lot of followers and many preferred tweets that he makes will have a greater node value compared to users who are few followers and few who like to tweet of those users.

Calculation of node values on twitter social media is formulated as follows.

$$nv_{(i)} = fr_{(i)} + \sum_{j}^{n} (fc_{(i,j)} + rt_{(i,j)}) \tag{9}$$

Where:

- $nv_{(i)}$ = *node value* from user *i*
- $fr_{(i)}$ = followers from user *i*
- $fg_{(i)}$ = following from user *i*
- $fc_{(i,j)}$ = *favorite* count for *tweet j* from user *i*
- $rt_{(i,j)}$ = *retweet* count for *tweet j* from user *i*
- n = count of all *tweet* and *quoted tweet* from user *i*

E. Modified Eigenvector Centrality

So far, we have created 4 calculation with 6 formulas. Those calculation will used to calculate modified eigenvector centrality. To see where the calculation attached is shown by Fig 3. The influence value, activeness value, and node value is attached to user node, and weight and influence value user is attached on edge of user node.

After we have all calculation based on dynamics that occur in social media, then we formulate the modified eigenvector centrality value. The calculation is based on eigenvector centrality by using adjacency matrix. After that, we then calculate buzzer value by using modified eigenvector centrality, influence, and activeness value. We used influence and activeness to show if the user is potential for becoming buzzer or not, because buzzer must active and have influence to other users.

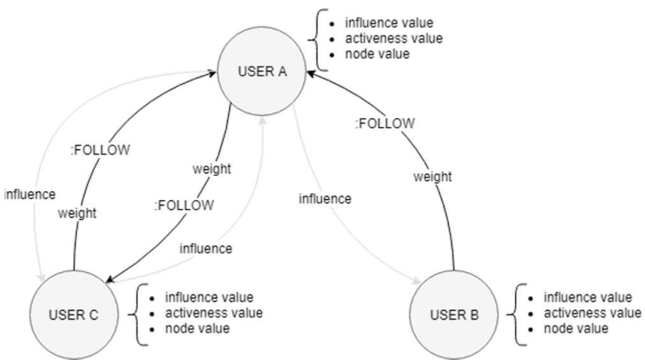

Fig 3 Calculation positon

The modified eigenvector centrality and buzzer value calculation then formulated as follows.

$$mec = (A_{aw} + A_{ivu}).nv \quad (10)$$

$$bv_i = mec_i.av_i.iv_i \quad (11)$$

Where:

- mec = *modified eigenvector centrality* of all user in matrix
- A_{aw} = *adjacency* matrix from edge adjusted weight
- A_{ivu} = *adjacency* matrix from *influence value user*
- nv = matrix of node value
- bv_i = *buzzer value* on user node *i*
- mec_i = value of modified eigenvector *centrality* on user node *i*
- av_i = *activeness value* from user node *i*
- iv_i = *influence value* from user node *i*

V. Implementation And Evaluation

It has been succeeded to implement the calculation for buzzer value that develop in environment python programming language and neo4j database. After implement the modified eigenvector centrality and buzzer value calculation, next step is to evaluate the calculation that have been made.

Evaluation conducted with 2 case, evaluate the execution time between buzzer value and other centrality and evaluate the information spreading by the recommended buzzer from buzzer value and modified eigenvector centrality in compare to other calculation like degree centrality, closeness centrality, and betweenness centrality.

A. *Execution Time*

This evaluation conducted by count the execution time between buzzer value and other centrality calculation like degree centrality, closeness centrality, and betweenness centrality. Execution time count by using data that has 5 node user and 9 node tweet, 18 node user and 112 node tweet, and 3151 node user and 5664 node tweet. The execution time shown as below.

Table 1 Execution time

Node	bc	cc	dc	bv	ec
5	1.4904	0.8165	1.183 3	2.1849	0.3378
18	46.471 9	1.7694	1.199 7	6.8239	0.5258
3151	NA	788.672 0	1.714 9	279.287 8	23.304 1

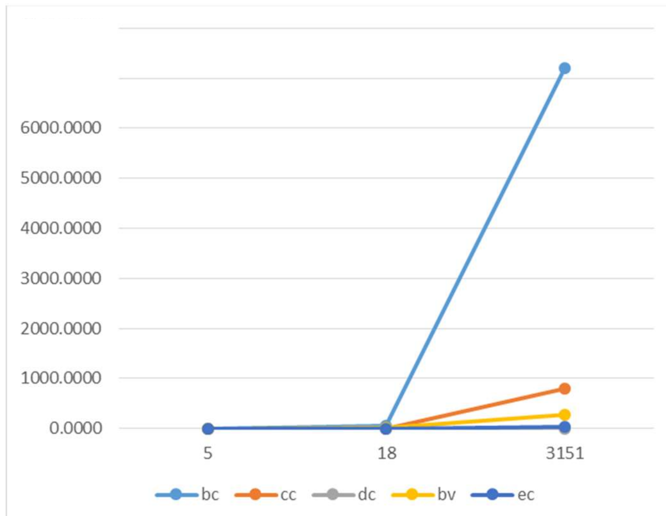

Fig 4 Execution time chart

From this data, we can see that dc (degree centrality) is the fastest one, because it only count the follower from user node. Eigenvector centrality take the second place, buzzer value take the third place, closeness centrality take the fourth place, and betweenness centrality is the worst one. Betweenness centrality is take a very long time to calculate many nodes.

Buzzer value is faster than closeness centrality when use 3151 node, but still below degree centrality and eigenvector centrality. It's acceptable because other calculation only calculate user node while buzzer value not only calculate user node but alto include tweet node. Buzzer value execution depend on how many user node and tweet node are. For the biggest one data that used, other centrality only calculate 3151 node, while buzzer value calculate 3151 user node + 5664 tweet node.

B. *Information Spreading*

This evaluation is conducted by testing spreading information from user node to other with considering influence value. We make threshold to influence value to filter which node can get the information. If the influence value among user node exceed the threshold, then we consider the information can reach that node, if the influence value not exceed the threshold then we assume that the information can't reach the node.

Data that used to test had been gathered from twitter API with initial seed "uxidbdg". The data contain 20169 users, 10525 follow relations, 57368 tweets, and total vertex is 77537. A small slice of data is shown by Fig *5*.

Fig 5 data test

5 big values of each centrality calculation, buzzer value, degree centrality, closeness centrality, and betweenness centrality is taken and run an information spreading test.

Table 2 Information spreading

Calculation	ID	D=0.75	D=0.5	D=0.25
Buzzer value	uxindonesia	1	10	10
	atnceu	1	1	2
	UXArmy	3	3	3
	afnizarnur	1	1	1
	aaprianil	2	2	2
		8	**17**	**18**
Degree centrality	TruthlnSociety	0	0	0
	ResultsPositive	0	0	0
	theUXSwitch	0	0	0
	KontakIndomaret	0	0	0
	mutialhanan	0	0	0
		0	**0**	**0**
Closeness centrality	uxindonesia	1	10	10
	nurizkaf96	1	1	1
	adamazkiya	2	4	4
	daengdoang	4	4	7
	icreativitor	0	0	0
		8	**19**	**22**
Eigenvector centrality	mutialhanan	0	0	0
	uxindonesia	1	10	10
	indonesia_ux	0	0	0
	afnizarnur	1	1	1
	daengdoang	4	4	7
		6	**15**	**18**

From Table 2 we can see the result from information spreading test. For the test, we use 3 threshold, 0.75, 0.5, and 0.25. With threshold 0.25 closeness centrality better than other, and buzzer value and eigenvector centrality got same value. With threshold 0.5 closeness centrality still better than other, and buzzer value got the second place. With threshold 0.75, bot closeness centrality and buzzer value got the same value with 8 reach node. From this result, we can see that information spreading with buzzer recommend from buzzer value can reach same node with closeness centrality when threshold value increase. This can happen because buzzer value calculate by using dynamics that occur in social media based on modified eigenvector centrality.

VI. Conclusion

Buzzer value calculation by using modified eigenvector centrality which include dynamics that occur in social media like interaction within users, level of activeness users, influence level of users, and value of its user node are more representing the real condition of social media. Buzzer value show buzzer depend on its influence and activeness value. Although a user has a big value of modified eigenvector centrality, the user not be considered as buzzer if the user not active or not has influence in social media graph that be observed. Execution time from buzzer value by using modified eigenvector centrality is better than betweenness centrality and closeness centrality, but still slower than degree centrality and eigenvector centrality. The spread of information using buzzer recommend from buzzer value by using modified eigenvector centrality got same value with closeness centrality when threshold increase, and better than other centrality calculation.

Future work from this research is to use sentiment value to make weight value in graph relation much better. With sentiment value, we can increase the weight if a user got positive comment or decrease the weight if a user got negative comment.

References

[1] Kietzmann, Jan H.; Kristopher Hermkens (2011). "Social media? Get serious! Understanding the functional building blocks of social media". *Business Horizons*. **54** (3): 241–251.

[2] Smith, K. (2016, Maret 7). *Marketing: 96 Amazing Social Media Statistics and Facts for 2016*. Dipetik Januari 24, 2017, dari Brandwatch: https://www.brandwatch.com/blog/96-amazing-social-media-statistics-and-facts-for-2016/

[3] Rouse, M. (2017, June). *What is social media analytics?* Retrieved from SearchBusiness Analytics: http://searchbusinessanalytics.techtarget.com/definition/social-media-analytics

[4] Hangal, S., MacLean, D., Lam, M. S., & Heer, J. (2010). All Friends are Not Equal: Using Weights in Social Graphs to Improve Search. *SNA-KDD Workshop.* Washington D.C: ACM.