Scalable Twitter User Clustering Approach Boosted by Personalized PageRank

Anup Naik(✉), Hideyuki Maeda, Vibhor Kanojia, and Sumio Fujita

Yahoo Japan Corporation, Tokyo, Japan
anaik@yahoo-corp.jp, hidmaeda@yahoo-corp.jp, vkanojia@yahoo-corp.jp, sufujita@yahoo-corp.jp

Abstract. Twitter has been the focus of analysis in recent years due to various interesting and challenging problems, one of them being Clustering of its Users based on their interests. For graphs, there are many clustering approaches which look at either the structure or at its contents. However, when we consider real world data such as Twitter Data, structural approaches may produce many different user clusters with similar interests. Similarly, content-based clustering approaches on Twitter Data produce inferior results due limited length of Tweet and due to lots of garbled data. Hence, these approaches cannot be directly used for practical applications. In this paper, we have made an effort to cluster Twitter Users based on their interest, looking at both the structure of the graph generated using Twitter Data, as well as its contents. By combining these approaches, we improve our results compared to the existing techniques, thereby generating results befitting the practical applications.

1 Introduction

There is huge fan following for idol groups or celebrities on social networks like Twitter. These fans frequently tweet about the events of the concerned celebrity, their latest news, videos, photos and other information; in a sense act as a *groupie* of the celebrity/idol groups. Such users can be used as a source to obtain real-time information about the concerned celebrity. This inspires us to cluster these *Social Influencers* in the fan following social network communities.

Huge user base has made Twitter user-graph very complex, and hence, analysis of Twitter Data has become burdensome. Existing structural approaches fail to perform effectively when we consider millions of nodes having active, inactive and spam users. On the other hand, content based approaches deteriorate because of limited length of textual contents in a tweet and garbled data. This observation acted as the basis of our approach to use both the structural as well as the content aspects of Twitter, thereby nullifying the drawbacks of each. Although follower list is available in Twitter, most of the users are inactive and interaction among them is lacking. The users in the follower list mostly read the tweets about the celebrity rather than posting something. Also, some of the celebrities (especially in Japan) do not have official accounts to get the follower list from. So for getting clusters, just taking the follower list is not effective.

J. Kim et al. (Eds.): PAKDD 2017, Part I, LNAI 10234, pp. 472–485, 2017.
DOI: 10.1007/978-3-319-57454-7_37

Our standpoint is consistent with the analyses made by Cha et al. where they endorsed *the million follower fallacy* by collecting empirical evidences [8]. Our contribution in this paper includes:

- We proposed a new approach for user clustering based on both content and graph features with topical relevance and influential ranking(using Personalized PageRank score) which can be used in many areas such as online advertising, viral marketing, personalized content dissemination and so on.
- We intensively compared our approach with content based, graph based and hybrid approaches in view of topical relevance and influential measures.
- Upon empirical evaluations, we confirmed that our approach outperforms strong and the state-of-the-art baseline systems even on massive data sets. Our data consisted of one month Twitter Data (1.6 TB in its compressed form).

2 Related Work

Graph Clustering: One of the graph clustering algorithm is *SCAN* [1]. It clusters vertices based on a structural similarity measure. It uses the fact that nodes in a clusters are densely connected with other nodes in the group and sparsely connected to nodes outside. Apart from clusters, it also finds *hub* nodes, which bridge two clusters, and *outlier* nodes which are vertices marginally connected to clusters. SCAN algorithm (Sect. 3.1) gives good results when we consider the structure of the graph and hence, a modified version of it acts as the first phase of our approach. One of the drawbacks of this algorithm is that it does not look at the contents of the nodes. So some of the clusters produced may belong to the same topic but stay as different clusters in this approach which we would like to merge in ideal case. Other algorithms use number of possible "betweenness" measures to iteratively remove edges to find clusters, as used in [3]. The *min-max cut method* [2] partitions the graph into two clusters A and B, by removing the minimum number of edges needed to isolate A and B. One drawback of this approach is that one has to specify the number of clusters beforehand. The most crucial problem is that if one cuts out a single node, one may achieve the optimum solution. In practice, this approach requires some constraint, such as $|A| \approx |B|$ which are inappropriate in real social networks.

Content Clustering: Content Clustering algorithms use various features of Twitter data to cluster users, such as the approach used in paper [5]. In this approach, they have used various similarity measure as feature for k-means clustering algorithm. Even though the paper claims to successfully cluster the users based on their interest, we are unable to reproduce the results due to large size of Twitter data we use, which is much larger than the data of the Twitter public API they used, that returns only a small portion of vast Tweet data. We empirically experienced that their method is not scalable to real Tweet streams. Hayashi et al. tried to detect hijacked topics when factorizing the user-term frequency matrix [10]. Our approach solves the same problem quite differently.

Social Graph Analysis: Leskovec et al. proposed *network community profile plot* to illustrate structural properties of network communities mainly based on the *graph conductance* measure [4]. They analyzed various kinds of social and information networks and pointed out the existence of many small scale and tight communities. We have addressed the same problem from a different but practically very effective approach. Cha et al. analyzed the Twitter network introducing three influential measures of users namely indegrees, retweets and mentions [8]. Their observation endorsed the intuitions of some observers such as [9] and pointed out that each measure indicate different features of tweets and users. Their influential measure is mainly for scientific observation and analysis purposes especially for topical relevant influentials and they eliminated chronologically mature topics due to hijacked keywords by spammers. Although we are inspired by their insights on the characteristics of Twitter network, they did not propose any methods to extract topically relevant influential users from real Twitter network in view of industry usage. Weng et al. proposed an approach for finding topic-sensitive influential twitterers [11]. They evaluated only on a small dataset of less than 5000 active users (compared to 3 million in our approach). Their method works only on a toy size dataset although the LDA method on Twitter texts normally outputs junk topics due to the topic hijacking and Twitter specific text usage such as short text with many emoticons.

3 Graph Clustering

First we describe the graph clustering algorithm which partitions twitter user networks by analyzing the structural properties of the graph. We adopt SCAN algorithm [1] and propose its enhancement, namely, Weighted SCAN, which is intended to more effectively partition users according to their network activities.

3.1 SCAN Algorithm

In this section, we describe in detail, the SCAN algorithm. This algorithm acts as the first phase in our approach. It takes two parameters as input; ϵ**:** a threshold value to determine structural similarity between two nodes and μ**:** the minimal size of a cluster. Let us define some of the commonly used terms. The list of symbols and its meaning is given in Table 1.

Definition 1. *STRUCTURAL SIMILARITY: The structural similarity between node u and v, denoted by* $\sigma(u, v)$*, is defined as*

$$\sigma(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{|\Gamma(u)||\Gamma(v)|}} \tag{1}$$

where Γ *is defined as* $\Gamma(v) = \{w \in V \mid (v, w) \in E\} \cup \{v\}$.

So, $|\Gamma(u) \cap \Gamma(v)|$ *becomes the number of common neighbours between u and v and* $\sqrt{|\Gamma(u)||\Gamma(v)|}$ *becomes the geometric mean of the two neighbourhoods' size.*

Definition 2. *CORE: Node u is core iff $|N_\epsilon[u]| \geq \mu$, where N_ϵ, called ϵ-neighbourhood, is $N_\epsilon[u] = \{v \in N[u] : \sigma(u, v) \geq \epsilon\}$.*

Definition 3. *HUB AND OUTLIER: Assume node u does not belong to any cluster C. $u \in H$ iff node v and w exist in $N[u]$ such that $C[v] \neq C[u]$. Otherwise $u \in O$.*

For more information about the algorithm, please refer the paper [1].

We have modified the structural similarity formula in SCAN algorithm to incorporate the weighted edge and named the new formula as weighted structural similarity and the algorithm as Weighted SCAN (WSCAN) algorithm, the details of which is described in Sect. 3.2.

Table 1. Terms and symbols used

Symbol	Definition	Symbol	Definition
ϵ	Threshold of structural similarity, $0 \leq \epsilon \leq 1$	H	Set of hubs in G
		μ	Minimal number of nodes in a cluster
O	Set of outliers in G	$N[u]$	Set of nodes in the structure neighbourhood of node u
G	Given graph	$N_\epsilon[u]$	Set of nodes in the ϵ-neighbourhood of node u
V	Set of nodes in G	$C[u]$	Set of nodes that belong to the same cluster as node u
E	Set of edges in G	$\sigma(u, v)$	Structural similarity between node u and v

Algorithm 1. SCAN Clustering Algorithm

```
Input  : Graph G(V, E), Parameters - ε, μ
Output: Set of clusters, hubs, and outliers
for each unclassified vertex v belongs to V do
    if Core(V) then
        Create new clusterID
        for all structurally similar neighbors x of V do
            if x is unclassified or non-member then
                Assign clusterID to x
            end if
            if x is also a core then
                Expand the graph using x also
            end if
        end for
    end if
end for
Further classify non-members into hubs and outliers
```

3.2 WSCAN as an Expansion of SCAN

In our approach, we construct a graph using the Reply and Re-Tweet (RT) features of tweet data. Adopting the original SCAN approach, an edge between two user nodes is created if a user has Re-Tweeted/Replied to another user, irrespective of how many times they did. However, some users RT more than

once, the tweets of the same user. In the same way, some users might reply to another user many times. Clearly influential users get more than one RT.

To take these features into consideration, we have modified the structural similarity formula of SCAN algorithm to use weighted structural similarity (σ_w).

Definition 4. *WEIGHTED STRUCTURAL SIMILARITY: Let $u, v \in V$ where V is the set of nodes, where each node represents a user, $\omega_{u,v}$ is defined as a linear function of RT and Reply count between the two nodes u and v.*

$$\sigma_w(u,v) = \frac{\omega_{(u,v)}}{\omega_{(u,v)} + 1} \frac{\mid \Gamma(u) \cap \Gamma(v) \mid}{\sqrt{|\Gamma(u) \mid\mid \Gamma(v)|}} \tag{2}$$

where Γ is defined earlier and $\omega_{u,v} = \alpha \cdot RT + \beta \cdot R$

This weight factor allows us to retain a significant node which would, otherwise, have been marked as a hub as described in [1]. Figure 2 shows a histogram of Reply and RT count. From this graph we observed that Reply and RT counts are following the same trend, and are proportional to each other. Hence we have used $\alpha = 1$ and $\beta = 1$ in Eq. 2 for all the evaluation purposes.

We illustrate the effectiveness of weights using a toy example Graph in Fig. 1. The edge weights represent Reply and RT counts, which are considered only in WSCAN and have no significance in SCAN algorithm. Let us consider two nodes: Node 5 and 6. In case of SCAN, both these nodes are structurally similar as they have same similarity (σ) to neighbours, i.e., $\sigma(2,5) = \sigma(4,6) = 0.63$, whereas in case of WSCAN, the similarities are $\sigma_w(2,5) = 0.32$, and $\sigma_w(4,6) = 0.47$ respectively. Node 6 seems to have stronger connection with the cluster of Nodes 1, 2, 3 and 4 due to larger edge weight, thereby acting as an influential user who actually has a great influence upon a member of a strongly connected community. By appropriately selecting the value of ϵ, Node 5 can be classified as an outlier, i.e. an insignificant node, whereas, Node 6 can be included in the cluster. The steps followed for clustering the graph in WSCAN are same as SCAN, explained in Algorithm 1.

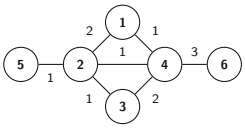

Fig. 1. Example of WSCAN (weighted structural similarity)

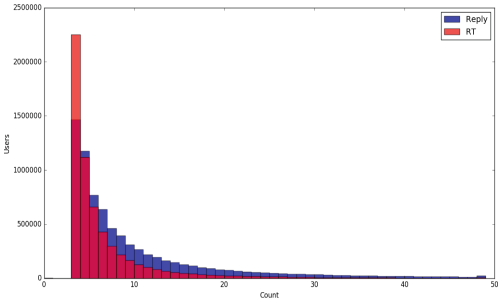

Fig. 2. Histogram of Reply and RT counts

4 Our Approach

Our approach has basically the following steps which are discussed in detail in subsequent sections. System diagram is shown in Fig. 3:

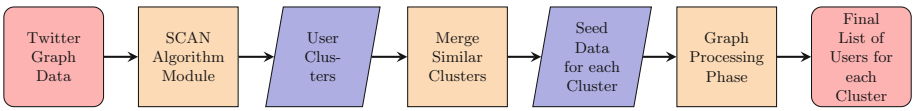

Fig. 3. System block diagram

- **Construction Phase:** Construct graph using Reply and RT feature.
- **Structural Clustering Phase:** Cluster the graph using WSCAN algorithm (refer Sect. 3.2).
- **Merging Similar Clusters:** Combine similar clusters and get the users of each cluster to be used as seed data for next step.
- **Graph Processing Phase:** Expand the list of seed users and rank them using Personalized PageRank algorithm.

4.1 Construction of Graph

Our approach starts with constructing an undirected, weighted graph from Twitter Data using Reply (R) and RT, since these are logically more meaningful than just follow feature when we consider the similarity of two users in interests. Consider two users u_1 and u_2. If u_1 has replied or re-tweeted the tweets of u_2, then (u_1, u_2) will have an edge in our graph. The sum of Reply and RT count between u_1 and u_2 becomes the weight of the edge, and is denoted by $\omega_{(u_1,u_2)}$. We have experimented with R, RT and both R+RT for constructing the graph. The results obtained after executing WSCAN on these three types of graphs with $\epsilon = 0.5, 0.45$ and 0.4 is summarized in Table 2.

Comparison of Reply (R), RT and Reply (R)+RT: Let us consider Table 2 and compare 3 types of graphs:

- **Reply Graph (R):** Users are represented by nodes and edge weight represents the reply count between two users.
- **Re-tweet Graph (RT):** Users are represented by nodes and edge weight represents the RT count between two users.
- **Reply and Re-tweet Graph (R+RT):** Users are represented by nodes and edge weight represents the sum of RT and reply count between two users.

For all the three graphs, we have considered edges whose weights are greater than 2. The observation is given below:

Table 2. Comparison of WSCAN output for the 3 types of graphs ($\mu = 2$)

ϵ	Type	#Vertices	#Hubs	#Outliers	#Clustered vertices	#W-SCAN clusters
0.5	R	2,472,492	1,250,077	402,691	819,724	340,960
	R+RT	2,596,565	1,559,828	383,581	653,156	277,340
	RT	1,121,444	649,780	292,281	179,383	78,114
0.45	R	2,472,492	1,045,536	349,296	1,077,660	410,364
	R+RT	**2,596,565**	**1,371,585**	**341,061**	**883,919**	**347,476**
	RT	1,121,444	610,051	281,454	229,939	94,720
0.4	R	2,472,492	818,927	286,833	1,366,732	457,001
	R+RT	2,596,565	1,149,285	289,819	1,157,461	406,682
	RT	1,121,444	562,266	267,904	291,274	111,709

- **Fraction of Hubs:** R+RT (0.60) > RT (0.57) > R (0.50): Consider a user U who has re-tweeted tweets of users from Cluster 1 and replied to tweets of users from Cluster 2. So in R+RT graph, U could be marked as a hub because of its involvement in both the groups. Whereas if we consider R and RT graphs, this user will be a part of Cluster 1 and Cluster 2 respectively.
- **Fraction of Outliers:** RT (0.26) > R (0.16) > R+RT (0.14): Outliers are those users which do not have any affiliation to any cluster. Nodes with very few edges to any cluster are marked as outliers. As users in R+RT graph have higher degree, the fraction of users marked as outliers are also less.
- **Fraction of WSCAN output Users:** R (0.33) > RT (0.16) > R+RT (0.10): The decrease in the fraction of clustered users in R+RT graph is because of the fact that a large fraction of them were marked as hub. This ensures that whatever users are left have strong affiliation to the cluster.

Another disadvantage of using just R or RT Graph is that the graphs produced are very sparse. When we visualized these graphs, there were many isolated clusters with just two nodes connected to each other. This depicts that there is a lot of one to one communication between pairs of nodes. So we have used R+RT Graph; R Graph to include users which are closer to each other in real-life, and RT Graph to include users which share similar content.

4.2 Find Clusters Using WSCAN

We input the graph constructed in Subsect. 4.1 to the WSCAN approach, as discussed in Sect. 3.2. This program produces clusters of users using weighted structural similarity measure as given in equation (2). We have used $\epsilon = 0.45$ and $\mu = 2$ (refer to the bold text in Table 2). This is because when we consider $\epsilon = 0.5$ and $\mu = 2$, we get very few users in clusters, as most of the users are filtered out as hubs and outliers. This leaves very few (poor quantity), but densely connected users (fine quality). On the other hand, when we consider $\epsilon = 0.4$ and $\mu = 2$, we get relatively good amount of users (fine quantity) but these users are sparsely connected (poor quality). So in order to deal with the tradeoff between quality and quantity, we have chosen $\epsilon = 0.45$ and $\mu = 2$.

4.3 Combine Similar Clusters

In this phase, we merge those clusters which are topically similar. For this we extract the HashTags of all the users in a cluster and combine them to make a document. We make document for each of the cluster produced from previous phase. Then we use Single Pass Clustering algorithm to cluster these document (which are basically clusters).

4.4 Expanding the Clusters (Graph Processing)

We use the output of each of the clusters produced from Subsect. 4.3 separately as seed data for the Graph Processing phase. For a given cluster, the Graph Processing phase basically calculates the Personalized PageRank (PPR) score [7] of all the seeds (in that cluster) and its connected nodes. It uses the Reply and RT features of Tweets to construct the graph, using which the PPR Score of the seeds and their connected nodes is calculated. We use the top 3000 nodes as per PPR Score for evaluating our result. This phase expands the nodes in a cluster by finding its topic related nodes which increases the coverage, removes most of the non-influential users and produces overlapping clusters.

5 Evaluation Experiments

In this section we describe the experiments conducted and their results. For all our experiments, we have used the Japanese Twitter data for the month of December 2015, size of which is about 1.6 TB in compressed form. From this data, we extracted the top 3,000,000 active users (using PageRank score). We have mostly used Hadoop, Pig, Java and Python in our implementations. The graphs in this paper have been generated in Python using graph-tool.[1]

5.1 Visualizing the Process and Observing the Effects

We have generated fan clusters for four celebrities/idol groups: *Arashi*[2], *AKB*[3], *Hanyu*[4] and *Yamashita*[5] using our approach.

Let us consider cluster *Arashi*. The output obtained after executing WSCAN and Single Pass Clustering is pictorially shown in Fig. 4. Here square shaped nodes represent the Seed nodes obtained after WSCAN and Single Pass Clustering. Different colours of the square nodes represent that they belong to different clusters. These different coloured square nodes are combined to a single cluster of Seed nodes (users) in the Single Pass Clustering Phase (Sect. 4.3). Figure 5 visualizes the output obtained after executing the Personalized PageRank

[1] https://graph-tool.skewed.de/.
[2] Arashi: A Japanese idol group.
[3] AKB (AKB48): A Japanese idol girls group.
[4] Hanyu (Yuzuru Hanyu): Japanese figure skater & 2014 Olympic champion.
[5] Yamashita (Tomohisa Yamashita): Japanese actor, singer, and TV host.

Table 3. Impact of single pass clustering

	Arashi	AKB	Hanyu	Yamashita
Clusters merged	9	10	2	4
Seed users obtained	70	79	138	109

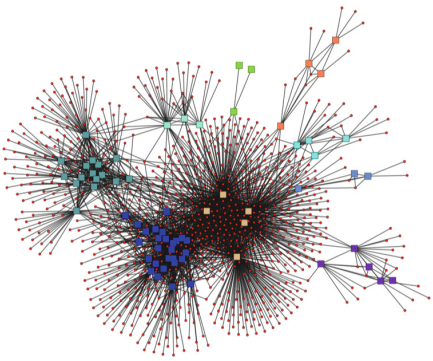

Fig. 4. Arashi cluster after executing WSCAN and Single Pass Clustering. (Color figure online)

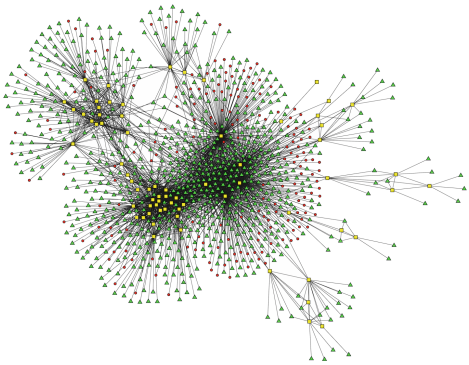

Fig. 5. Arashi cluster after executing Personalized PageRank (Graph Processing). (Color figure online)

(Graph Processing Phase) on the seed nodes obtained after the Single Pass Clustering phase for the cluster *Arashi*. The yellow coloured nodes represent the seed nodes. The green coloured nodes represent the nodes obtained after the execution of Personalized PageRank. The red coloured nodes represent the neighbours of seed nodes which are not in the output of Graph Processing Phase. Table 3 shows the number of clusters merged by the Single Pass Clustering Phase. As seen from the table, for *Arashi*, 9 clusters (shown by different coloured square nodes in Fig. 4) were combined by the Single Pass Clustering Phase to produce 70 seed nodes. Details of other clusters can also be seen in the table.

5.2 Evaluation Design

The main problem with evaluation of the clusters is that the perfect set of users for any of these clusters is unknown. So we use crowdsourcing for evaluating our results. Crowdsourcing also eliminates biasing of test results.

Crowdsourcing: For Crowdsourcing, we frame the question so as to check whether the given user is actually interested in the group or not. We have used the topics *Arashi*, *AKB*, *Hanyu* and *Yamashita* because of the fact that these are famous in Japan and it would be easy to do crowdsourcing. 844 people participated in crowdsourcing. Each question was reviewed by three people. The average of opinion of three people was taken as the answer of a question. A sample crowdsourcing question with its options is given in Table 4.

Table 4. Sample crowdsourcing question

Question: Is this user <https://twitter.com/twitter_screen_name> interested in Arashi?			
1. Yes, this user is interested	2. No, this user is not interested	3. I don't know	4. Cannot access the account

We are using the output of the Graph Processing Phase for evaluation, considering the top 3000 users based on Personalized PageRank score. We are evaluating our approach with the following approaches:

- **Normal SCAN algorithm (NS):** This system uses SCAN [1]. We use the graph constructed using Reply and RT features (same as in our approach) of Twitter Data as input. We have used $\epsilon = 0.6$ (equivalent to $\epsilon = 0.45$ in WSCAN when we consider minimum edge weight equal to 3) and $\mu = 2$.
- **RB clustering (RB):** This system uses the HashTag information and its TF-IDF as input. We extract HashTags and tokenize them to create document for each user. So considering all the users, we have a list of documents. Then we calculate the TF-IDF of the HashTags. This acts as the feature in RB (Repeated Bisection) based clustering algorithm. We use Bayon[6] Clustering Tool for doing this and extract the users of concerned cluster for evaluation.
- **RB Clustering followed by Personalized PageRank (RB-PR):** This system is similar as the RB system. Only difference is that, using the nodes in the cluster (obtained from RB clustering) as seeds, we expand the cluster, calculating the PPR Score for the seed nodes and their connected nodes (this step finds more nodes related to the seeds). We then consider the top 3000 nodes using the PPR Score.
- **Normal SCAN followed by content clustering (NS-C):** This system is a combination of Normal SCAN (NS) and Content Clustering. Here, we use output of the NS system (described above) and perform Single Pass Clustering technique (except that the input is SCAN clusters and not WSCAN clusters).

5.3 Evaluation Metrics

User Influence Weighted Discounted Cumulative Gain: Discounted cumulative gain (DCG) is a measure of the quality of ranking of information items such as documents and used to evaluate the ranking effectiveness of, for example, the search engines [6]. We extended this in order that the measure takes user's "influenceability" into consideration because influential users are more important in our task. We defined User Influence Weighted DCG (UIWDCG) score to calculate the influence and topical relevance of users in our results. For this, we have used top 3000 users obtained after Graph Processing phase for four topics. UIWDCG score for a list of top n users is defined as:

[6] https://code.google.com/archive/p/bayon/.

$$UIWDCG(n) = g_1 \cdot log(IL(v_1) + 1) + \sum_{i=2}^{n} \frac{g_i \cdot log(IL(v_i) + 1)}{log(i)} \quad (3)$$

where $g_i \in [0, 1, 2, 3]$, $IL(v)$ is the number of in-links to the vertex v representing a Twitter user. g_i is the evaluation score for vertex v at rank i, representing the number of workers out of three who marked this user as relevant to the considered topic, as per crowdsourcing results.

Table 5. Precision (correct seed nodes/total seed nodes) of seed nodes

	Arashi	AKB	Hanyu	Yamashita
Our approach	0.89(62/70)	0.82(65/79)	0.75(103/138)	0.86(94/109)
RB	0.75(758/1013)	0.45(36/80)	0.62(205/331)	0.90(101/111)

5.4 Results and Discussions

Figure 6 shows the average UIWDCG score of all the approaches. X-axis shows the Number of Users considered from top while calculating UIWDCG score. Y-axis shows the Average UIWDCG score. We have used 3 matrices to calculated the UIDWCG score, Favorite Count, Mention Count and Reply+RT count. The value of $IL(v)$ varies depending upon the matrix selected. It is clear from the graph that when we consider top 3000 users, our approach gives more influential users than other approaches. Table 6 shows the UIWDCG score of top 100, 500, 1000 and 3000 users of each cluster for various systems considered for evaluation. We observed the following points:

- Pure graph based approach, such as **NS**, is very weak, and even the least performing one. It failed to output more than 1000 users for any topic.
- Pure content based approach **RB** is also inadequate and it is not comparable with our system.
- The WSCAN step proves to be crucial one as it removes the users who either do not belong to any topic group or the ones who belong to many groups.
- As seen in Fig. 6, although **RB-PR** is better in the beginning, our approach outperforms after rank 300, finally 18.3% gain is observed at rank 3000.
- The quality of seed nodes produced is very good in our approach (ref Table 5).
- Figure 6 shows that RB has better performance than RB-PR. We conclude that good quality seeds are imperative for the good performance of Personalized Pagerank. The seeds obtained from Hashtag frequency in RB-PR were mediocre and hence, deteriorated the performance of Pagrerank. On the other hand, seeds obtained from WSCAN were superior and hence enabled Pagerank to perform effectively.

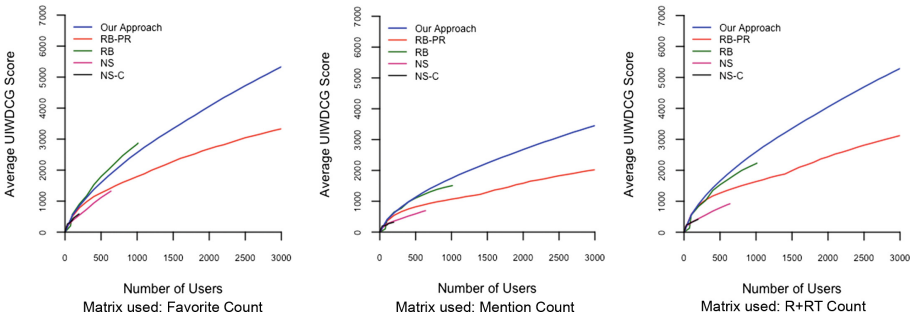

Fig. 6. Average UIWDCG (of Arashi, AKB48, Hanyu, Yamashita) vs #Users

Table 6. UIWDCG score of the four clusters that we considered.

Technique	Top 100	Top 500	Top 1000	Top 3000
Arashi				
Our approach (G+C+G)	575	**1832**	**3092**	**6865**
RB-PR (C+G)	**639**	1658	2545	6006 -
NS-C (G+C)	274	413(194)[a]	-	-
RB (C)	628	1532	2214	-
NS (G)	318	787	-	-
AKB				
Our approach (G+C+G)	473	1564	**2554**	**5981**
RB-PR (C+G)	**654**	**1619**	2439	5333
NS-C (G+C)	193(52)[a]	-	-	-
RB (C)	129(80)[a]	-	-	-
NS (G)	46(15)[a]	-	-	-
Hanyu				
Our approach (G+C+G)	605	**1807**	**2983**	**6160**
RB-PR (C+G)	**612**	1455	2451	5223
NS-C (G+C)	238(34)[a]	-	-	-
RB (C)	543	-	-	-
NS (G)	272(83)[a]	-	-	-
Yamashita				
Our approach (G+C+G)	**578**	**1556**	**2282**	**4283**
RB-PR (C+G)	575	1263	1635	3121
NS-C (G+C)	420	444(118)[a]	-	-
RB (C)	518	-	-	-
NS (G)	192(24)[a]	-	-	-

[a]Since 100/500 nodes are unavailable, number of nodes given inside () are used for calculation.

6 Conclusions

We have proposed a scalable method to cluster Twitter Users based on their interest looking at both the structure as well as the contents. According to our empirical experiments using large Twitter Data, both pure structural and content-based clustering approaches failed to gather thoroughly, the users with certain topical interests. We have introduced the notion of constructing the graph using Reply and RT features. We have modified SCAN [1] to incorporate Reply and RT count as edge weights (WSCAN), the benefits of which were explained in Sect. 3.2. The parameters of WSCAN were chosen so as to obtain few, but influential seed data, as seen in Table 5. In order to deal with isolated clusters having similar contents, we have used content-based merging using Textual Similarity. We have illustrated the effects of this step by visualizing the graph data (Figs. 4 and 5). The superiority of the proposed process to merge clusters obtained by WSCAN algorithm is observed in Table 3. The Graph Processing phase improved the coverage of our system and enabled us to obtain influential users related to the seed data. We carried out evaluations for topical relevance of clustered Twitter users by Crowdsourcing and observed significant improvement over state-of-the-art approaches on both precision-recall curves and UIWDCG measures. Our system outperforms the best performing baseline system, RB-PR, with 18.3% gain in Average UIWDCG Score for Top 3000 Users, as seen in Fig. 6. Possible future work includes looking into the contents in more detail to improve the results. Also, Topic Recognition approach needs to be improved for much better results.

References

1. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.: Scan: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 824–833. ACM (2007)
2. Ding, C.H., He, X., Zha, H., Gu, M., Simon, H.D.: A min-max cut algorithm for graph partitioning and data clustering. In: 2001 Proceedings of the IEEE International Conference on Data Mining, ICDM 2001, pp. 107–114. IEEE (2001)
3. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
4. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: Proceedings of the 17th International Conference on World Wide Web, pp. 695–704. ACM (2008)
5. Zhang, Y., Wu, Y., Yang, Q.: Community discovery in Twitter based on user interests. J. Comput. Inf. Syst. **8**(3), 991–1000 (2012)
6. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. (TOIS) **20**(4), 422–446 (2002)
7. Haveliwala, T.: Topic-sensitive pagerank. In: Proceedings of the 11th International Conference on World Wide Web, Honolulu, Hawaii, USA, pp. 517–526 (2002)
8. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, P.K.: Measuring user influence in Twitter: the million follower fallacy. In: ICWSM, vol. 10, pp. 10–17 (2010)

9. Avnit, A.: The million followers fallacy. http://blog.pravdam.com/the-million-followers-fallacy-guest-post-by-adi-avnit/ (2009). Accessed 2 Aug 2016
10. Hayashi, K., Maehara, T., Toyoda, M., Kawarabayashi, K.-I.: Real-time top-r topic detection on Twitter with topic hijack filtering. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2015, pp. 417–426 (2015)
11. Weng, J., Lim, E.-P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM 2010, pp. 261–270 (2010)