Yandex

Twitter graph case study

Data

Due to Twitter's new Terms of Services, we cannot share data containing
(for more info, read RWW's article "How Recent Changes to Twitter's Te
Research")

Social graph

- **Download**
  * Now we offer direct download links: [tar.gz] [zip]
  ~~twitter_rv.tar.gz.torrent (34KB) or twitter_rv.zip.torrent (26KB) (#~~
  twitter_rv.tar.gz, 6,475,352,982 bytes, MD5: c31b4c2d6f3ae325e5
  twitter_rv.zip, 4,859,337,443 bytes, MD5: 5f2399aac71c604ac5a1
  ---
  twitter_rv.net, 26,172,280,241 bytes, MD5: 9c0f7983a523edd1b75
- Format

```
USER \t FOLLOWER \n
```

  * USER and FOLLOWER are represented by numeric ID (integer)
  * These numeric IDs are the same as numeric IDs Twitter managed
  * Therefore, you can access a profile of user 12 via http://api.twitte
  * For details, see Twitter API Page
- Example

```
12    13
12    14
12    15
16    17
```

  * Users 13, 14 and 15 are followers of user 12.
  * User 17 is a follower of user 16.

http://an.kaist.ac.kr/traces/WWW2010.html

What is Twitter, a Social Network or a News Media?

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon
Department of Computer Science, KAIST
335 Gwahangno, Yuseong-gu, Daejeon, Korea
{haewoon, chlee, hosung}@an.kaist.ac.kr, sbmoon@kaist.edu

ABSTRACT

Twitter, a microblogging service less than three years old, commands more than 41 million users as of July 2009 and is growing fast. Twitter users tweet about any topic within the 140-character limit and follow others to receive their tweets. The goal of this paper is to study the topological characteristics of Twitter and its power as a new medium of information sharing.

We have crawled the entire Twitter site and obtained 41.7 million user profiles, 1.47 billion social relations, 4,262 trending topics, and 106 million tweets. In its follower-following topology analysis we have found a non-power-law follower distribution, a short effective diameter, and low reciprocity, which all mark a deviation from known characteristics of human social networks [28]. In order to identify influentials on Twitter, we have ranked users by the number of followers and by PageRank and found two rankings to be similar. Ranking by retweets differs from the previous two rankings, indicating a gap in influence inferred from the number of followers and that from the popularity of one's tweets. We have analyzed the tweets of top trending topics and reported on their temporal behavior and user participation. We have classified the trending topics based on the active period and the tweets and show that the majority (over 85%) of topics are headline news or persistent news in nature. A closer look at retweets reveals that any retweeted tweet is to reach an average of 1,000 users no matter what the number of followers is of the original tweet. Once retweeted, a tweet gets retweeted almost instantly on next hops, signifying fast diffusion of information after the 1st retweet.

To the best of our knowledge this work is the first quantitative study on the entire Twittersphere and information diffusion on it.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and behavioral sciences

General Terms

Human Factors, Measurement

Keywords

Twitter, Online social network, Reciprocity, Homophily, Degree of separation, Retweet, Information diffusion, Influential, PageRank

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

1. INTRODUCTION

Twitter, a microblogging service, has emerged as a new medium in spotlight through recent happenings, such as an American student jailed in Egypt and the US Airways plane crash on the Hudson river. Twitter users follow others or are followed. Unlike on most online social networking sites, such as Facebook or MySpace, the relationship of following and being followed requires no reciprocation. A user can follow any other user, and the user being followed need not follow back. Being a follower on Twitter means that the user receives all the messages (called *tweets*) from those the user follows. Common practice of responding to a tweet has evolved into well-defined markup culture: RT stands for retweet, '@' followed by a user identifier address the user, and '#' followed by a word represents a hashtag. This well-defined markup vocabulary combined with a strict limit of 140 characters per posting conveniences users with brevity in expression. The *retweet* mechanism empowers users to spread information of their choice beyond the reach of the original tweet's followers.

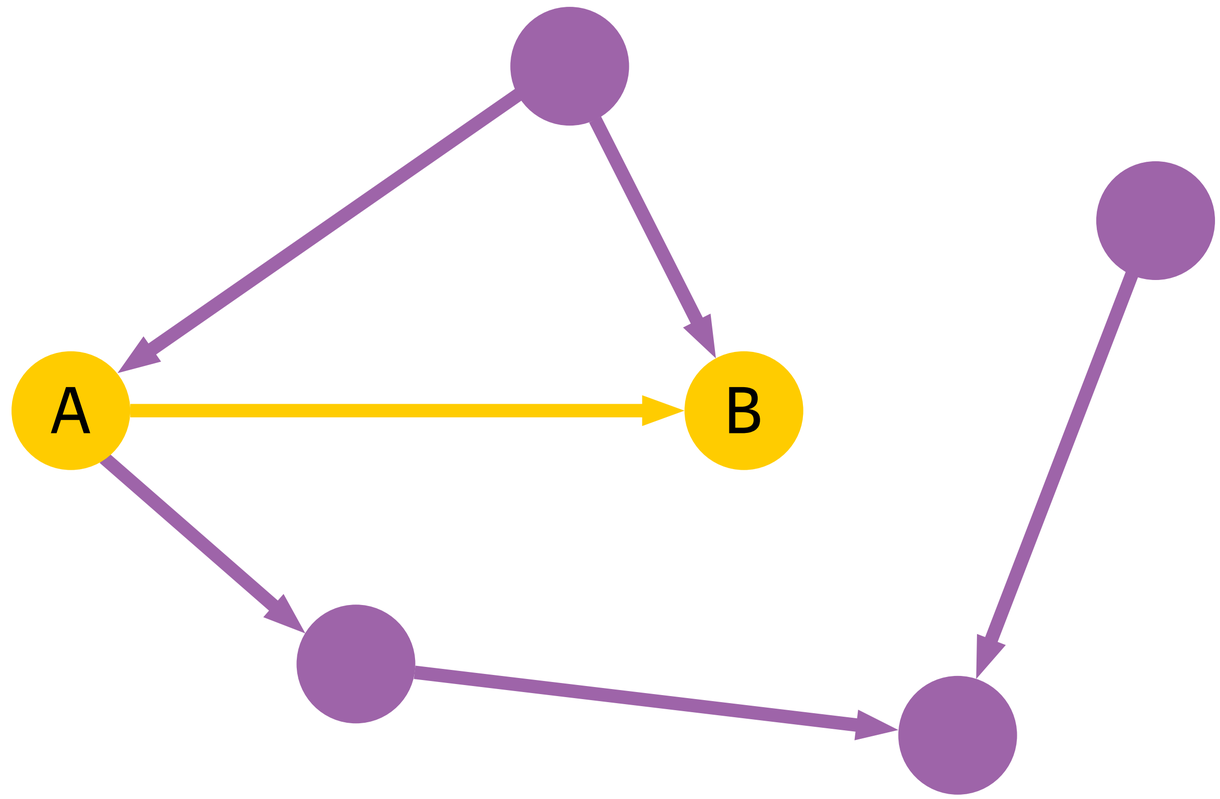
How are people connected on Twitter? Who are the most influential people? What do people talk about? How does information diffuse via retweet? The goal of this work is to study the topological characteristics of Twitter and its power as a new medium of information sharing. We have crawled 41.7 million user profiles, 1.47 billion social relations, and 106 million tweets[1]. We begin with the network analysis and study the distributions of followers and followings, the relation between followers and tweets, reciprocity, degrees of separation, and homophily. Next we rank users by the number of followers, PageRank, and the number of retweets and present quantitative comparison among them. The ranking by retweets pushes those with fewer than a million followers on top of those with more than a million followers. Through our trending topic analysis we show what categories trending topics are classified into, how long they last, and how many users participate. Finally, we study the information diffusion by retweet. We construct retweet trees and examine their temporal and spatial characteristics. To the best of our knowledge this work is the first quantitative study on the entire Twittersphere and information diffusion on it.

This paper is organized as follows. Section 2 describes our data crawling methodology on Twitter's user profile, trending topics, and tweet messages. We conduct basic topological analysis of the Twitter network in Section 3. In Section 4 we apply the PageRank algorithm on the Twitter network and compare its outcome against ranking by retweets. In Section 5 we study how their popularity rises and falls among users over time. In Section 6 we focus information diffusion through retweet trees. Section 7 covers related work and puts our work in perspective. In Section 8 we conclude.

[1]We make our dataset publicly available online at: http://an.kaist.ac.kr/traces/WWW2010.html

Social graph

- Vertexes — users (41'652'230)
- Edges – "follows" relation (1'468'365'182)
- 6G compressed, 25G uncompressed
- Edge (A, B) =
  = User A follows B =
  = User B is followed by A

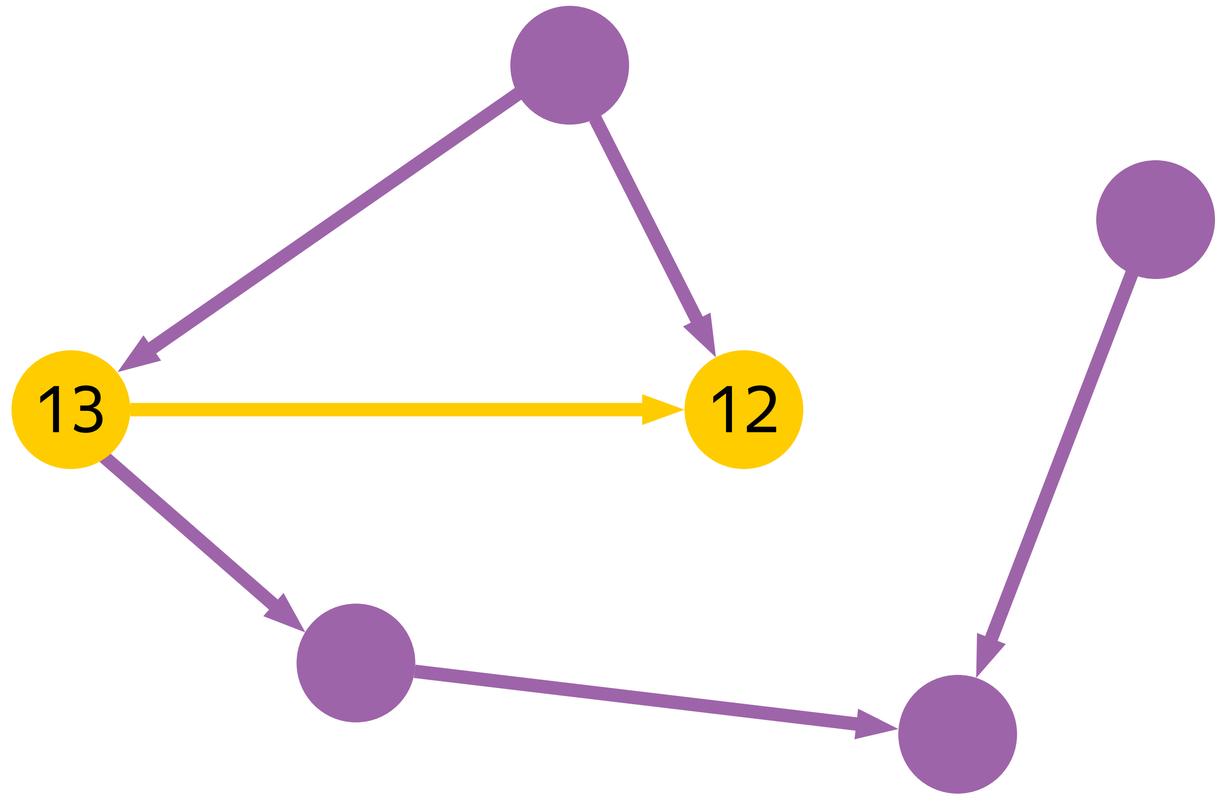


Social graph

› Format:
one edge per line
USER \t FOLLOWER \n
USER & FOLLOWER
– numeric ids

› Sample 1:

```
12   13   #  edge   (13, 12)
12   14   #  edge   (14, 12)
12   15
16   17
...
```

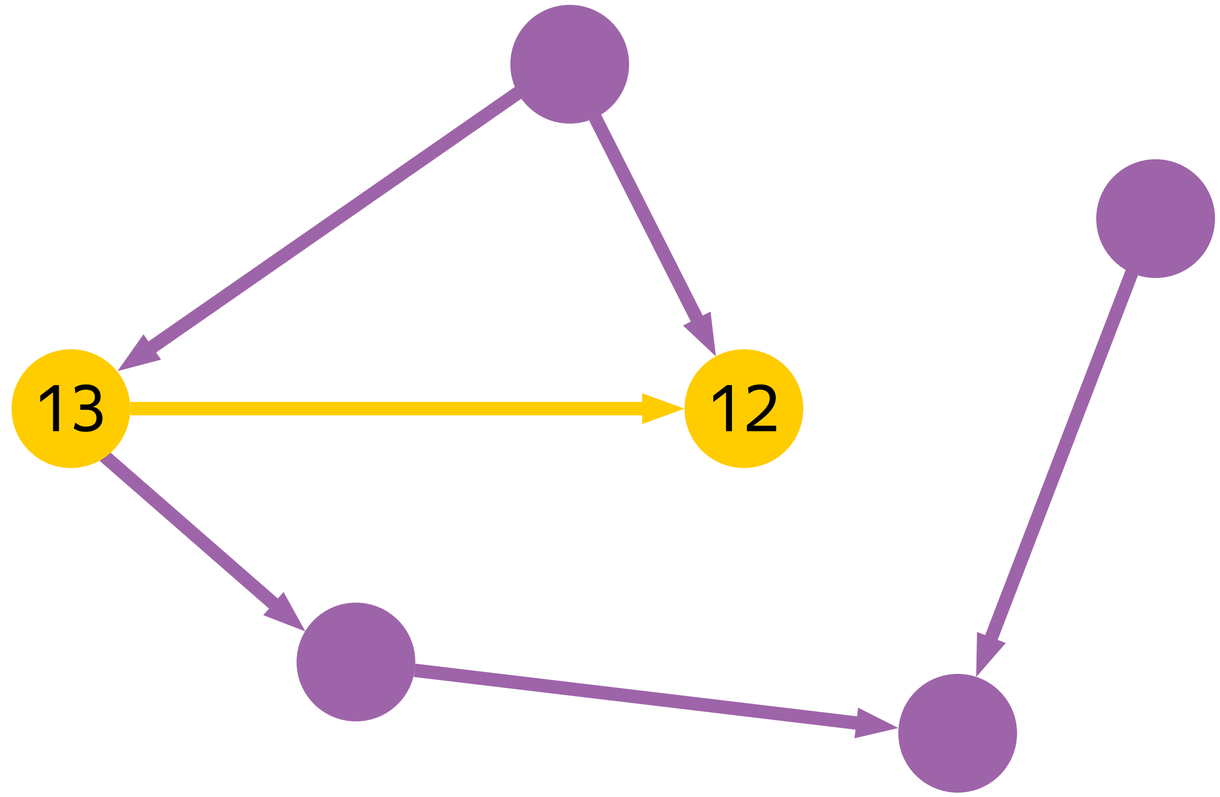


Social graph

› Format:
one edge per line
USER \t FOLLOWER \n
USER & FOLLOWER
– numeric ids

› Sample 2
(all edges incident to 4825):

2790	4825
4825	2790
4825	10316422
4825	55041391
4825	59104182



```
$ pyspark
...
>>>
```

```
$ pyspark
...
>>> raw_data = sc.textFile(“hdfs:///data/twitter/twitter_rv.net”)
>>>
```

$ pyspark

```
...
>>> raw_data = sc.textFile(“hdfs:///data/twitter/twitter_rv.net”)
>>> def parse_edge(s):
...     user, follower = s.split(“\t”)
...     return (int(user), int(follower))
...
>>> edges = raw_data.map(parse_edge).cache()
>>>
```

$ pyspark

```
...
>>> raw_data = sc.textFile(“hdfs:///data/twitter/twitter_rv.net”)
>>> def parse_edge(s):
...   user, follower = s.split(“\t”)
...   return (int(user), int(follower))
...
>>> edges = raw_data.map(parse_edge).cache()
>>> # find users with the largest number of followers
```

Feel free to pause the video here to come up with your own solution

$ pyspark

```
...
>>> raw_data = sc.textFile(“hdfs:///data/twitter/twitter_rv.net”)
>>> def parse_edge(s):
...   user, follower = s.split(“\t”)
...   return (int(user), int(follower))
...
>>> edges = raw_data.map(parse_edge).cache()
>>> # find users with the largest number of followers
>>> import operator
>>> follower_counts = edges.mapValues(lambda v: 1).reduceByKey(operator.add)
>>> follower_counts.top(5, operator.itemgetter(1))
[(19058681, 2997469), (15846407, 2679639), (16409683, 2674874),
(428333, 2450749), (19397785, 1994926)]
>>>
```

$ pyspark

```
...
>>> raw_data = sc.textFile(“hdfs:///data/twitter/twitter_rv.net”)
>>> def parse_edge(s):
...   user, follower = s.split(“\t”)
...   return (int(user), int(follower))
...
>>> edges = raw_data.map(parse_edge).cache()
>>> # find users with the largest number of followers
>>> import operator
>>> follower_counts = edges.mapValues(lambda v: 1).reduceByKey(operator.add)
>>> follower_counts.top(5, operator.itemgetter(1))
[(19058681, 2997469), (15846407, 2679639), (16409683, 2674874),
(428333, 2450749), (19397785, 1994926)]
>>>
```

$ pyspark

```
...
>>> raw_data = sc.textFile(“hdfs:///data/twitter/twitter_rv.net”)
>>> def parse_edge(s):
...   user, follower = s.split(“\t”)
...   return (int(user), int(follower))
...
>>> edges = raw_data.map(parse_edge).cache()
>>> # find users with the largest number of followers
>>> import operator
>>> follower_counts = edges.aggregateByKey(0, lambda a, x: a + 1,
operator.iadd)
>>> follower_counts.top(5, operator.itemgetter(1))
[(19058681, 2997469), (15846407, 2679639), (16409683, 2674874),
(428333, 2450749), (19397785, 1994926)]
>>>
```

$ pyspark

```
...
>>> raw_data = sc.textFile(“hdfs:///data/twitter/twitter_rv.net”)
>>> def parse_edge(s):
...   user, follower = s.split(“\t”)
...   return (int(user), int(follower))
...
>>> edges = raw_data.map(parse_edge).cache()
>>> # find users with the largest number of followers
>>> import operator
>>> follower_counts = edges.aggregateByKey(0, lambda a, x: a + 1,
operator.iadd)
>>> follower_counts.top(5, operator.itemgetter(1))
[(19058681, 2997469), (15846407, 2679639), (16409683, 2674874),
(428333, 2450749), (19397785, 1994926)]
>>> follower_counts.top(5, lambda p: -p[1]) [(22020180, 1), (25165920, 1),
(25690275, 1), (21496020, 1), (30933240, 1)]
>>>
```

$ pyspark

You have learned:

> how to load data from HDFS

> how to use the 'aggregateByKey' transform

> that social graphs exhibit extreme skewness

BigDATAteam