Prediction of User Retweets Based on Social Neighborhood Information and Topic Modelling

Pablo Gabriel Celayes and Martín Ariel Domínguez(✉)

Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba, Córdoba, Argentina
mdoming@famaf.unc.edu.ar

Abstract. Twitter and other social networks have become a fundamental source of information and a powerful tool to spread ideas and opinions. A crucial step in understanding the mechanisms that drive information diffusion in Twitter, is to study the influence of the social neighborhood of a user in the construction of her retweeting preferences. In particular, to what extent can the preferences of a user be predicted given the preferences of her neighborhood.

We build our own sample graph of Twitter users and study the problem of predicting retweets from a given user based on the retweeting behavior occurring in her second-degree social neighborhood (followed and followed-by-followed). We manage to train and evaluate user-centered binary classification models that predict retweets with an average $F1$ score of 87.6%, based purely on social information, that is, without analyzing the content of the tweets.

For users getting low scores with such models (on a tuning dataset), we improve the results by adding features extracted from the content of tweets. To do so, we apply a Natural Language Processing (NLP) pipeline including a Twitter-specific adaptation of the Latent Dirichlet Allocation (LDA) probabilistic topic model.

Keywords: Retweet prediction · Social model
Social network analysis · Machine learning · LDA · SVM

1 Introduction

In the last years, social media sites (e.g., Twitter, Facebook, and YouTube) have become increasingly massive. The Twitter application is an online real-time social and information network that enables its users to post, read and share messages of up to 140 characters, known as tweets. Every time a user writes a tweet, Twitter attaches to it a unique identifier and a creation timestamp. Another frequently used function on the bird net is the "retweet", which is the action of reposting someone else's tweet inside your own message stream (the "timeline").

F. Castro et al. (Eds.): MICAI 2017, LNAI 10633, pp. 146–157, 2018.
https://doi.org/10.1007/978-3-030-02840-4_12

This work addresses the central question of determining the influence of a user's social environment on her retweeting behavior. To this end, we train and evaluate classifier models that seek to predict retweets by a given user, considering the retweeting behavior of her second degree social neighborhood (followed users, and followed by followed). Additionally, we explore the possibility of improving these purely social models using Natural Language Processing (NLP) techniques to include content-based features.

The present work has been carried out in the following phases:

- Construction of a dataset of Twitter users, followers and tweets.
- Study of models to learn and predict retweeting preferences on this dataset, based on information about the social neighborhood of each user.
- Study of possible improvements to social prediction models, introducing Natural Language Processing techniques.

The work constitutes not only an interesting analysis of different algorithms and techniques, but also a way of understanding how users are influenced by their social environment.

The rest of this paper continues as follows: In Sect. 2 we compare our work with some relevant works in the area. Next, we provide a description of how we built the dataset for our experiments, both for purely social experiments and for those where we consider the content of tweets. In Sect. 4, we describe how we built the social model for retweet prediction, and also how we added content-based features to it: first, using the LDA probabilistic topic model [1], second, using the Twitter LDA variation proposed in [13]. Section 5 contains the analysis of the results obtained. We finish with Sect. 6, including conclusions and possible lines of research for future works.

2 Related Work

Over the last years, the topic of user recommendation in Twitter has been widely studied. Some examples of this interesting research topic are [3–5,7,10]. In particular, in [3], authors defined a measure between users called "Similar-to" and their framework focuses on discovering top similar users for each type of user in Twitter. Summarizing, these proposed works [3–5,7] show the recommender system currently implemented by Twitter to suggest new users to follow. They have ranked the relationships between users by using different techniques based on features, such as users' retweets, favorite tweets, email address and some historical data. In contrast, our approach is trying to predict a retweet, based on the structure of followed users (first and second degree followed) and the content of tweets.

Another interesting research topic in this area aims to predict if a given tweet will become a viral one, that is, trying to measure its "retweetability". This approach is very related to ours as it attempts to achieve this goal by predicting if a tweet will be retweeted. Some relevant works along these lines are [8,9,11,12]. In [8] the authors base their prediction in the content of tweets extracted from

CHOUDHURY-EXT dataset [2]. They use different features based on the text of the tweet, such as topics extracted by the LDA algorithm. They train a Logistic Regression model, and get to the following conclusion about Twitter users: As a general rule, a tweet is likely to be retweeted when it is about a general, public topic instead of a narrow, personal one. For instance, a tweet is unlikely to be retweeted when it is addressed to another Twitter user directly, while their topic analysis revealed that general topics affecting many users (e.g.: a general election or Christmas) are more likely to be retweeted.

In [9], the authors predict retweets in a dataset crawled using the Twitter Streaming API throughout October 2010. They adopt a machine learning approach based on the passive-aggressive algorithm, using social features such as the author of the tweet, the number of followers, friends, statuses, favorites, among others. They also used features related to the content of the tweet itself: number of hashtags, mentions, URLs, trending words, the length of the tweet, novelty, etc. They built a general model of prediction, and their best model obtained a 46.6% of F_1 score in average.

In contrast to the two works mentioned earlier, our approach to retweet prediction generates a different model for each given user. Also, while previous models employ social data in aggregated ways or combined with other kind of features, our initial models are based purely on specific social information about which neighbor users retweeted each given tweet. This makes our approach a more direct way of assessing the influence of social neighborhood behaviour in the retweet behavior of users.

This first approach achieves an average F_1 score of 88%. In cases where the pure social models performed poorly on a tuning dataset; we incorporated content-related features, obtained by including LDA topics from the tweets.

It is important to mention that although in [2] they study a more global concept of retweetability than the user-centered one we explore, our approach can be extended to have more general notions of retweetability as we explain in the last section.

3 Dataset

In this section, we describe how we build the datasets used in this work. First, we explain how we build the social graph used in the social based prediction. Second, we describe how we build the tweets dataset.

3.1 Social Graph

To the purpose of this work, we wanted to build a network where each user has a rich enough neighborhood of users, which would allow us to build social models for any user in the network.

To this end, we decide to build a homogenous network, trying to ensure that all users have similarly sized neighborhoods, by means of the following process:

First we built a large enough *universe graph* from which a homogenous subgraph would be subsequently sampled. The universe graph is built as follows: starting with a singleton graph containing just the account of a specific Twitter user $\mathcal{U}_0 = \{u_0\}$, we perform 3 iterations of the following procedure: (1) Fetch all users followed by users in $\mathcal{U}_i$; (2) From that group, filter only those having at least 40 followers and following at least 40 accounts; (3) Add filtered users and their edges to get an extended $\mathcal{U}_{i+1}$ graph.

From this process, we obtain a universe graph $\mathcal{U} := \mathcal{U}_3$ with $2,926,181$ vertices and $10,144,158$ edges.

Since we want to fetch shared content for every user of our social graph and we also want it to be homogenous (note that users added in the last step will have no outgoing edges), we take a subgraph following this procedure:

- We start off with a small sample of seed users S, consisting of users in $\mathcal{U}$ having out-degree 50.
- For each of those, we add their 50 most socially affine followed users. The affinity between two users is measured as the ratio between the number of users followed by both and number of users followed by at least one of them.
- We repeat the last step for each newly added user until there are no more new users to add.

This procedure returns a graph $\mathcal{G}$ with $5,180$ vertices and $229,553$ edges. We call it the homogeneous K-degree closure ($K = 50$ in this case) of S in the universe graph $\mathcal{U}$.

3.2 Content

For each user in the graph $\mathcal{G}$ from the previous section, we fetched their timelines (i.e. all tweets written or shared) for one month, from August 25th until September 24th, 2015. Finally, we only kept the tweets written in the Spanish language –using the Twitter API tag for filtering–, resulting in a set $\mathcal{T}$ of $1,636,480$ tweets. We do so to be able to analyze the content of the tweets with LDA which we do only for Spanish.

4 Experimental Setup

In this work we aim to build models capable of predicting the retweeting preferences of a given user, based both on what we know of her social environment and the content she previously shared. This section describes how we build those models attempting to achieve this over a selection of users and tweets from the $(\mathcal{G}, \mathcal{T})$ dataset defined before. We start describing the predictive model based only on the social environment. Then, we analyze the possibility of improving these predictions using NLP techniques, which also take into account the text of the tweets.

4.1 Social Media Graph Information

The main focus of this work is to predict if a given user u will share a given *tweet* based on information on who in user u's neighborhood has shared it. Since the process of feature extraction, modeling, and parameter tuning is computationally expensive, these experiments are performed on a selected subset of users. We begin by describing the criterion with which this subset was selected. We then describe how we generate, for each user u, a neighborhood of users E_u and a set T_u of potentially interesting *tweets*. Then, we describe the feature extraction process based on T_u and the partitioning into sets of *training*, *tuning*, and *evaluation*. Finally, we explain the process of training classifiers and tuning their parameters.

User Selection Process. The process of training classifier models is computationally expensive, so we decided to focus on a subset of selected users. It is desirable to work with users having enough shared content and also a rich enough level of social interactions. We took the 1000 users with highest Katz centrality [6] in $\mathcal{G}$, and on the other side, we picked the 1000 users with the highest number of retweets. We restrict our analysis to users belonging to both lists, which leaves us a set U of 194 users, with an average number of 494 retweets per user.

It is important to remark that in our experiments the universe of users is still $\mathcal{G}$, with all its users and connections. $\mathcal{G}$ is used to generate the environments of each user in U whose retweeting preferences we are trying to predict.

Visible Tweets. Using the Twitter API we do not have explicit information about whether or not a user saw a given tweet, but we can at least take a universe of *potentially viewed* tweets. This is simply the set of all the tweets written or shared by the users followed by u.

We exclude from this set those tweets *written* by u herself, since our focus is in recognizing interesting external content, and not on studying the generation of content from a particular user. Formally this set is defined as:

$$T_u := \left(\bigcup_{x \in \{u\} \cup \texttt{followed}(u)} \texttt{timeline}(x) \right) - \{t \in T | \texttt{author}(t) = u\}, \qquad (1)$$

where $\texttt{followed}(u) := \{x \in G | (u, x) \in \texttt{follow}\}$ and $\texttt{timeline}(x)$ is the set of all tweets written or shared by x for tweets fetched in T.

For some users, the set T_u turned out to be too large, making the process of experimenting and model training too computationally intensive. We decided to prune each T_u to a maximum of $10,000$ tweets. Since the retweeted tweets are the minority class, we decided to keep all positive examples and do the pruning by subsampling on the class of negative examples (non-retweets).

User's Environment. As the user u can only see tweets shared by those users she follows, the information about her extended network can provide more indicators of the degree of interest of a tweet t. That is why we decided to take as a user's environment not only the users she follows, but also to continue one more step in the `follow` relation and include the users followed by them. This is, we take all users (other than u herself) to 1 or 2 steps forward from u in the directed graph G, formally:

$$E_u = \left(\bigcup_{x \in \{u\} \cup \texttt{followed}(u)} \texttt{followed}(x) \right) - \{u\} \tag{2}$$

Environment Features. Now, we can build the set of vectors needed for the predictive model centered in user u. Given $E_u = \{u_1, u_2, \ldots, u_n\}$, we define for each tweet $t \in T_u$ the following vector of boolean features:

$$v_u(t) := [\texttt{tweet\_in\_tl}(t, u_i)]_{i=1,\ldots,n}, \tag{3}$$

where $\texttt{tweet\_in\_tl}(t, u) := \begin{cases} 1 & t \in \texttt{timeline}(u) \\ 0 & \text{otherwise} \end{cases}$

Note that the content of tweet t is not considered, we only include the information about who retweeted t. Gathering the vectors of all tweets in $T_u = \{t_1,\ ldots, t_m\}$ into a single matrix, our vectorized dataset is constructed as:

$$M_u := [\texttt{tweet\_en\_tl}(t_i, u_j)]_{\substack{1 \le i \le m, \\ 1 \le j \le n}}, \tag{4}$$

where the variable to be predicted for each tweet t is $\texttt{tweet\_in\_tl}(t_i, u)$. Putting together all values of the target variable for user u, we obtain the following objective vector: $y_u := [\texttt{tweet\_in\_tl}(t_i, u)]_{1 \le i \le m}$.

Splitting the Dataset. As usual, to evaluate the performance of our models in unseen data, we separate a portion of the dataset for evaluation that won't be used by the algorithms in the training phase. On the other hand, in our process of extending models with additional features, we will need to make decisions based on the quality of the first family of models, but we don't want those decisions to influence the final evaluation. This leads us to create an additional partition of dataset, taking a subset that we call the *tuning* dataset. Summarizing, we decide, for every user u to randomly split M_u in training (M_u^{tr}), tuning (M_u^{tu}) and evaluation (M_u^{te}) datasets, containing 70%, 10% and 20% of M_u dataset respectively. We denote the corresponding output labels for each of these datasets y_u^{tr}, y_u^{tu} and y_u^{te}.

4.2 Adding Natural Language for Prediction

In this section, we present how we add information about the content of the tweet in the classifiers. We describe the process to transform the text into feature

vectors. Then, we enumerate all stages in the transformation: the normalization and cleaning of the text, its tokenization, generation of *bag of words* feature vectors and the reduction of their dimensionality using LDA topic models.

Selection of Users. In this section we focus on improving the prediction quality for those users whose social model performs poorly. To this end, we take all users who have an $F1$ score lower than 0.75 in the tuning dataset M_u^{tu}. We are also interested in analyzing if there is any improvement in cases of better quality, so we extend the sample with a random selection of 10 more users.

This results in a U_{NLP} set of 37 users, over which we will try NLP extended models, while keeping the purely social models for the remaining 157 users in U.

Preprocessing. In this section we enumerate the sequential transformations performed to turn a tweet into a vector of numeric features describing its content.

- **Normalization.** In the first step, we remove the following for normalizing purposes: URLs, lowercase, accents, unusual characters, vowel repetitions (e.g.: turn `goooooal` into `goal`) and blank spaces repetitions.
- **Tokenization.** Next, we proceed to split the text into tokens by means of: punctuation removal, word splitting, *stopwords* removal, stemming and removal of single characters words. For this purpose, we use the `NLTK`[1] package, that implements stopword removal and stemming for Spanish language.
- **Bag of words.** We keep only tokens occurring in at least 100 tweets and at most 30% of the entire corpus of tweets. This results in a vocabulary $V = \{t_1, \ldots, t_{11238}\}$ of 11,238 terms, which we use to represent any tweet t as a vector of integers containing the number of occurrences of each term from V in t (the so-called *bag-of-words* representation):

$$v_{BOW}(tweet) := [count(t_i, tokens(tweet))]_{i=1}^{11238}, \tag{5}$$

  where $count(t, tokens)$ counts the occurrence of term t in the list $tokens$. In the case of short texts like tweets, most terms occur 0 or 1 times, so v_{BOW} can be regarded as a boolean vector.
- **LDA and TwitterLDA.** Training models with a large dimensionality leads to problems of both efficiency and overfitting. This is why we need to reduce bag-of-words vectors to a representation with fewer dimensions, but that still captures relevant information about the content of each tweet. To do so, we use the LDA model which discovers underlying topics within a given corpus and represents them as probability distributions of occurrence of terms. In turn, this algorithm can be applied to texts to model them as vectors of topical scores/probabilities.
  The short length of tweets and the fact that they normally cover just one main topic can lead to poor performance of classical LDA algorithm. That's

[1] http://www.nltk.org/.

why we also experiment with `TwitterLDA`[13], a variation of LDA that modifies the underlying probabilistic model to group tweets by user and assign a single topic to each tweet. For both `LDA` and `TwitterLDA`, we experiment with models of 10 and 20 topics.

5 Results

In this section, we describe how the retweet prediction models were generated and evaluated. We compare our models to simple baseline models that predict retweets for a user based only on popularity of tweets, given by the number of "likes"[2] and the number of "retweets" for a given tweet. To build the baseline model, we use simple feature vectors with the number of retweets and the number of likes. Then we train a Logistic Regression Classifier for each user in U.

5.1 Social Models

We analyze now the results obtained from training and evaluating user-centered classifier models using the feature vectors described in Sect. 4.1. For each user in U we trained an `SVC`[3] model from `scikit-learn`[4] on her training dataset (M_u^{tr}, y_u^{tr}), using the class `GridSearchCV` to perform a 3-fold cross-validated parameter search for the optimal configuration among all possible combinations of the following parameter choices:

```
{ "C":[0.01, 0.1, 1], "class_weight":["balanced", None],
"gamma":[ 0.1, 1, 10 ], "kernel":["rbf", "poly"]}
```

Finally, for each user u in U we evaluate the resulting classifier over the test set (M_u^{te}, y_u^{te}), obtaining an average $F1$-score of about 88% (Table 1), with a median score also around the same value. A more detailed analysis of the distribution of observed precision, recall and $F1$ scores over all users in U can be seen in Fig. 1.

Table 1. Performance of models over U on M_u^{te}.

Model	Avg. $F1$	Avg. Pr.	AVg. Rec.
Baseline	23.57%	21.1%	44.9%
Social	87.68%	97.4%	81.1%
Soc.+LDA(10)	85.37%	91.1%	80.9%
Soc.+LDA(20)	85.04%	92.1%	80.1%
Soc.+TW-LDA(10)	87.99%	98.1%	80.9%
Soc.+TW-LDA(20)	87.97%	97.9%	81.0%

Table 2. Performance evaluations over U_{NLP} on M_u^{te}.

Model	Avg. $F1$	Avg. Pr.	AVg. Rec.
Baseline	23.96%	20.4%	38.9%
Social	76.46%	95.6%	67.2%
Soc.+LDA(10)	64.38%	65.5%	66.0%
Soc.+LDA(20)	62.62%	66.4%	61.9%
Soc.+TW-LDA(10)	78.12%	97.9%	66.2%
Soc.+TW-LDA(20)	77.99%	97.0%	66.4%

[2] Likes are represented by a small heart and are used to show appreciation for a tweet. The number of "likes" is the number of the users which express it for a given tweet.

[3] For *Support Vector Classifier*, name of classical Support Vector Machines (SVM) in `scikit-learn`.

[4] http://scikit-learn.org/.

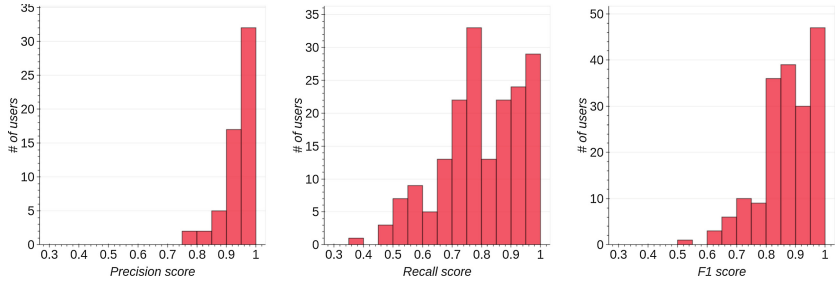

Fig. 1. Precision, recall and $F1$ score histograms of social model over $u \in U$ in M_u^{te}.

In Fig. 2 we compare the performance of our social prediction models with the baseline for users in U. We can see that the social models are in fact capturing the notion of social environment influence over each user, beyond what can be inferred from just looking at how popular a tweet is.

5.2 Social + NLP Models

In this section, we present some improvements on the purely social prediction by adding NLP features describing the content of tweets. We experiment with both classical LDA and TwitterLDA [13] models of 10 and 20 topics, on all selected 37 users in U_{NLP}. We use the same procedure as before to fit the classifiers, namely SVM models with hyperparameters adjusted through a 3-fold cross-validated

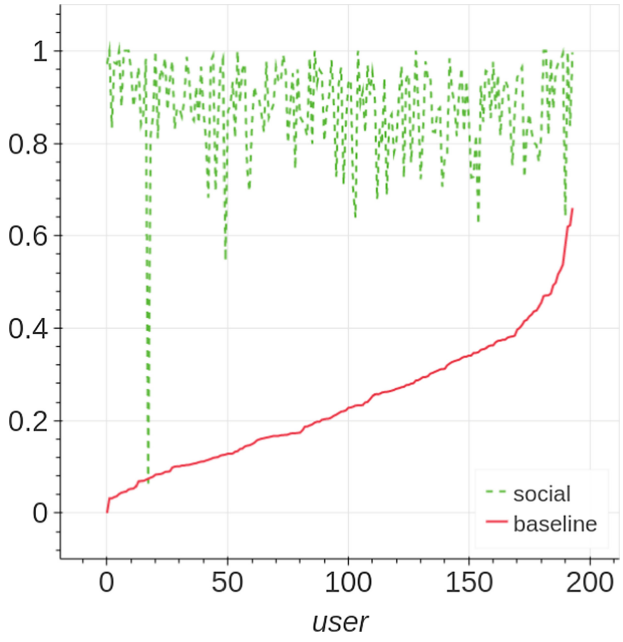

Fig. 2. $F1$ scores on baseline vs. social for $u \in U$ on M_u^{te}.

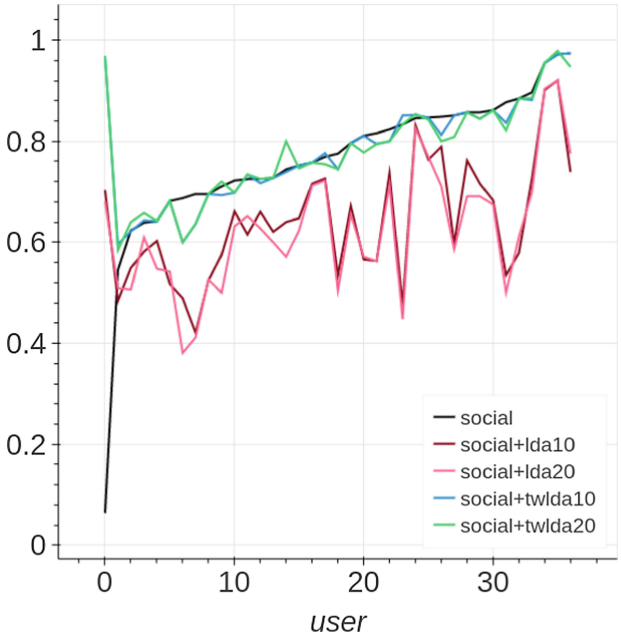

Fig. 3. $F1$ scores on soc vs. soc-LDA vs. soc-TwLDA for $u \in U_{NLP}$ on M_u^{te}.

search. To overcome convergence issues, in the case of classical LDA we apply scaling to all columns and impose a limit of 100,000 iterations to the underlying numeric optimization algorithm.

We compare results with the models obtained in the previous section and the baseline, by analyzing performance over the test sets (M_u^{te}, y_u^{te}). It is important to remark that these datasets are not used in the training phase or in the selection of users in U_{NLP}. The latter were picked based on their performance on the tuning sets (M_u^{tu}, y_u^{tu}). In Table 2 we can see the results obtained over test sets of each user, restricted to users in U_{NLP} for all models[5]. The best performing models are the ones that use `TwitterLDA` with 10 topics, attaining an average improvement of 1.7% over the purely social models.

From Table 2 we can also observe that classical `LDA` turns out to be very unsuitable for modeling Twitter content. Not only it doesn't attain an improvement, but it decreases the quality of models by more than a 12%, most likely due to overfitting on too descriptive topic probability features.

Only after switching to the `TwitterLDA` variation (which assigns a single topic to each tweet instead of a probability distribution over topics), we are able to obtain an improvement over the purely social model. These differences can be clearly observed in Fig. 3.

[5] We denote with `social+lda10` the models that combine social features and classical LDA features with 10 topics. Similar notation applies for 20 topics and the `TwitterLDA` variation.

6 Conclusions and Future Work

During the development of this work, we confirmed our idea that the analysis of social networks can provide very useful tools when implementing content recommendations, allowing us to also better understand the connections between the interests of a user and her social environment. We found it surprising to see the high performance of social environment-based predictions, without even taking the content into account. We also noticed that the extraction of topics with LDA, beyond its usefulness in tasks of corpus exploration and understanding, has enough potential to describe text content in a few dimensions of features. Using the TwitterLDA variation of the classical LDA model is of great importance at this point, and turned out to be the only way to achieve an improvement in the average prediction quality. We have many ideas to continue this work; we now continue to describe here the most relevant ones. One possible way is to try to infer a classifier capable of characterizing a user's retweeting behavior. This is, when the user is going to retweet, how much she is influenced by the social environment and how much she is by the content of the tweet. In the case of obtaining a classifier of this type, we could combine the social environment model and the content model in a better way. We also have in mind adding more features to our model such as number or rate of retweets among followed users, number or rate of retweets between followed by followed, number or rate of retweets between friendships (users who follow each other with the central user), among others. Finally, another interesting direction is to use our current user-centered retweet predictions to develop a notion of retweetability within groups or communities of users.

References

1. Blei, D.M.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
2. Choudhury, M.D., Lin, Y.R., Sundaram, H., Candan, K.S., Xie, L., Kelliher, A.: How does the data sampling strategy impact the discovery of information diffusion in social media? In: ICWSM. The AAAI Press (2010)
3. Goel, A., Sharma, A., Wang, D., Yin, Z.: Discovering similar users on twitter. In: In 11th Workshop on Mining and Learning with Graphs (2013)
4. Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D., Zadeh, R.: WTF: The who to follow service at twitter. In: Proceedings of the 22nd International Conference on World Wide Web. International World Wide Web Conferences Steering Committee (2013)
5. Kamath, K., Sharma, A., Wang, D., Yin, Z.: RealGraph: user interaction prediction at twitter. In: In User Engagement Optimization Workshop @ KDD (2014)
6. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
7. Lin, J., Kolcz., A.: Large-scale machine learning at twitter. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM (2012)
8. Nasir, N., Gottron, T., Kunegis, J., Alhadi, A.C.: Bad news travel fast: a content-based analysis of interestingness on twitter. In: WebSci 2011: Proceedings of the 3rd International Conference on Web Science (2011)

9. Petrovic, S., Osborne, M., Lavrenko, V.: RT to win! predicting message propagation in twitter. ICWSM **11**, 586–589 (2011)
10. Yanar, A.: Combining topology-based & content-based analysis for followee recommendation on Twitter. Ph.D. thesis, Middle East Technical University, April 2015
11. Zaman, T.R., Herbrich, R., Van Gael, J., Stern, D.: Predicting information spreading in twitter. In: Workshop on computational social science and the wisdom of crowds, NIPS, vol. 104, pp. 17599–17601. Citeseer (2010)
12. Zhang, Q., Gong, Y., Wu, J., Huang, H., Huang, X.: Retweet prediction with attention-based deep neural network. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM (2016)
13. Zhao, W.X., et al.: Comparing Twitter and Traditional Media Using Topic Models. In: Clough, P., et al. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 338–349. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20161-5_34