Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media

RT to Win! Predicting Message Propagation in Twitter

Saša Petrović
10 Crichton Street
Edinburgh EH8 9AB
United Kingdom
sasa.petrovic@ed.ac.uk

Miles Osborne
10 Crichton Street
Edinburgh EH8 9AB
United Kingdom
miles@inf.ed.ac.uk

Victor Lavrenko
10 Crichton Street
Edinburgh EH8 9AB
United Kingdom
vlavrenk@inf.ed.ac.uk

Abstract

Twitter is a very popular way for people to share information on a bewildering multitude of topics. Tweets are propagated using a variety of channels: by following users or lists, by searching or by retweeting. Of these vectors, retweeting is arguably the most effective, as it can potentially reach the most people, given its viral nature. A key task is predicting if a tweet will be retweeted, and solving this problem furthers our understanding of message propagation within large user communities. We carry out a human experiment on the task of deciding whether a tweet will be retweeted which shows that the task is possible, as human performance levels are much above chance. Using a machine learning approach based on the passive-aggressive algorithm, we are able to automatically predict retweets as well as humans. Analyzing the learned model, we find that performance is dominated by social features, but that tweet features add a substantial boost.

1 Introduction

Twitter is a microblogging service that allows users to post short (140 characters in length) messages, called *tweets*, which are then read by anyone who subscribed to receive the author's updates. Although a very popular means of communication with over 100 million users, many aspects of Twitter still remain poorly understood. Here we focus on the phenomenon of *retweeting*, or propagating other people's posts to one's followers. Understanding how retweeting works can provide insight into how information spreads through large user communities and also has applications in marketing.

We consider the following questions in this paper: i) is it at all possible to predict when something will be retweeted?, ii) can we build models which automatically predict retweets, and how well do these models perform?, and iii) what factors contribute most in predicting retweets (e.g., how well can we predict retweets without even reading the tweet)?

We first conduct a human experiment showing that the task is possible as humans perform significantly better than random chance. To automatically predict retweets, we use a machine learning approach based on the passive-aggressive algorithm. We adapt this algorithm to take tweet creation time into account, resulting in the best overall model. Finally, we take a look at how much different features contribute towards predicting a retweet and find that social features (especially number of followers and lists) perform very well, but that there is a substantial gain in using features of the tweet itself. A comparison of our best model with human performance shows that our approach does as well as humans on this task.

2 Related Work

A very in-depth study of the various aspects of retweeting was presented in (Boyd, Golder, and Lotan 2010). They explicitly interviewed Twitter users on the reasons why they retweet, and on what they retweet the most. While (Boyd, Golder, and Lotan 2010) provide an interesting insight into the practice of retweeting, they make no attempt to actually predict when something will be retweeted.

(Suh et al. 2010) conducted a large-scale analysis of factors that impact retweeting. They found that the number of followers and friends have a lot of impact, while, e.g., number of statuses and favorites do not. They also train a single-layer perceptron from a small subset of tweets, but do not use the model for actual prediction. Instead, they only examine the learned weights of the model. In this paper, we conduct a much more thorough investigation of the prediction task, while also putting emphasis on efficiency and being able to deploy our approach on live Twitter data. (Zaman et al. 2010) use a collaborative filtering approach to predict, for a pair of users, whether a tweet written by one will be retweeted by the other user. (Zaman et al. 2010) use a fairly poor feature set (IDs of both users, their number of followers, and the words in the tweet) and the task of pairwise predicting retweets makes this approach infeasible for a large-scale, especially streaming setting with millions of users.

Other studies of retweets concentrated on analyzing a small number of very popular tweets and their corresponding retweet networks (Nagarajan, Purohit, and Sheth 2010), or on predicting information diffusion by analyzing how tweets on the same topic spread (Yang and Counts 2010).

3 Retweeting

Retweeting is the action of reposting someone else's tweet inside your own message stream, and there are generally two ways to do it on Twitter. Users can either manually edit the original tweet and add "RT @userA" (or something similar) to indicate that the original tweet came from userA, or they can use a *retweet* button which does not allow them to change the original tweet. Due to problems with identifying the connection between the original tweet and the subsequent manual retweet (people can do this in any number of ways), we focus upon retweets made using the *retweet* button. In this case, the tweet-retweet connection is unambiguously marked in Twitter's API.

The prediction task is as follows. Tweets arrive one at a time, and for each one we want to predict whether someone will retweet it. There is one caveat here: Twitter only provides a small sample of the entire stream through their streaming API. As a result, it is possible that a tweet is retweeted, but for that retweet to not appear in the sample. When constructing our training/testing set, we will thus incorrectly label that tweet as not being retweeted. Unfortunately, there is no getting around this problem, so it should be kept in mind when reading the results.

Dataset. We run our experiments on a stream of tweets crawled from the Twitter streaming API[1] throughout October 2010. We gathered a total of roughly 21 million tweets. We split this set into a training and a test set using a 90-10 split such that the test set comprises the last 10% of data (over 2 million tweets) ordered by time. In total, there were over 24 million unique tokens in the data.

4 Predicting retweets

4.1 Streaming prediction

We now consider the task of automatically predicting retweets. The tweets are coming in as a stream of text, and we want to make the prediction as soon as we see the tweet and discard it right away – this is the realistic setting in which an actual system would be deployed. Such treatment of the task warrants the use of online learning algorithms, as opposed to traditional, batch ones which operate on the entire dataset. This is why we use the passive-aggressive (PA) algorithm of (Crammer et al. 2006). PA maintains a linear decision boundary and for each new example it tries to classify it correctly with a certain margin, while keeping the decision boundary as close as possible to the old one. The choice of the algorithm is not crucial and any approach that is feature-based would be appropriate. There are three variations of the PA algorithm, depending on the loss function used. In this paper we use what (Crammer et al. 2006) refer to as PA-II version – we tried all three versions of the update and PA-II performed the best. We set the aggressiveness parameter C of the PA algorithm to 0.01 in our experiments.

4.2 Time-sensitive modelling

Assuming that each time of day can have some specific rules as to what gets retweeted (maybe tweets containing the word "oil" are more retweeted in the morning than in the evening), we introduce separate, *local* models, trained only on a particular subset of data. We use a local model for every hour of the day, depending on when a tweet was written. This means that, in addition to one global model, which is trained on all the data, we also have 24 local models, each one trained only on those tweets written in a specific hour. Note that all these models are trained using the standard PA algorithm. When we have to make a prediction, we modify the original prediction rule of PA with the following:

$$\hat{y} = sgn(\langle w_g, x \rangle + \lambda \langle w_l, x \rangle), \tag{1}$$

where w_g is the global weight vector, w_l is the local weight vector for the specific hour the tweet was written in, and all the weight vectors are L_2 normalized. λ is a weight which corresponds to the proportion of training examples that the chosen local model has seen, i.e., $\lambda = n_l/N$, where n_l is the total number of training examples that the local model has seen, and N is the total number of training examples. λ encodes our confidence in the local classifier – the more examples it has seen, the more we trust its judgment. Note that we have tried, instead of fixing λ, using a stacked classifier trained on $\langle w_g, x \rangle$, $\langle w_l, x \rangle$, and λ as features, but it did not outperform a model with a fixed λ. We therefore opt to fix λ for the sake of speed and simplicity.

4.3 Features

We divide the features into two distinct sets: social features (features related to the author of the tweet), and tweet features (which encompass various statistics of the tweet itself, along with the actual text of the tweet).

Social features. We use the following features related to the author of the tweet: **number of followers, friends, statuses, favorites, number of times the user was listed, is the user verified, and is the user's language English**. Number of followers and friends has been consistently shown to be a good indicator of retweetability (Suh et al. 2010; Zaman et al. 2010), whereas the number of statuses and favorites was not found to have significant impact (Suh et al. 2010). *Lists* are a way to organize friends into groups according to some criteria (e.g., members of family, people who tweet about complexity in computer science, etc.). If a user is *listed* many times, i.e., many lists follow him, this should mean that he tweets about things that are interesting to a larger user population, and his tweets will reach a broader audience. *Verification* is used by Twitter mostly to confirm the authenticity of celebrity accounts. We found that 91% of tweets written by verified users are retweeted, compared with 6% for tweets where the author is not verified. This shows that almost anything that celebrities write will get retweeted, and thus having this feature should improve performance. Our prior analysis also showed that tweets written in English are more likely to be retweeted so we use a binary feature indicating if the user's language is English. We are not aware of any prior work that analyzes the effect of lists, verification, and language on retweetability.

Tweet features. We use the following features related to the tweet itself: **number of hashtags, mentions, URLs, trending words, length of the tweet, novelty, is the tweet a reply, and the actual words in the tweet**. Hashtags, URLs, and mentions were already shown by (Suh et al. 2010) to have a high correlation with retweetability. A reply indicates a direct message from one user to another, so intuitively it should make the tweet less likely to be retweeted, as it is not directed to a general audience. Trending topics are a set

[1] `http://stream.twitter.com/`

of possibly multi-word terms that are popular on Twitter at a given time. Including words from trending topics should make a tweet more likely to be read, as trending topics are often used as keywords for search. Novelty score is the cosine distance from the tweet to its nearest neighbor in vector space – this definition of novelty is common in the first story detection literature. This score tells us how much new content a tweet has, compared to all the tweets we have seen so far. We also use the actual words in the tweet as features. The words were lowercased and split on whitespace; we made no attempt to resolve the shortened URLs. To the best of our knowledge, no prior work on analyzed the effect of trending topics, novelty, length, and replies on retweetability.

5 Experiments

5.1 Human experiments

To make sure that the task of predicting retweets is at all possible, we first conduct an experiment with human subjects. We perform two experiments where we presented two human subjects with 202 pairs of tweets, and asked them to mark the one they think would get retweeted. In each pair exactly one of the tweets was retweeted. The order of the two tweets was chosen randomly for each pair. We evaluate the performance as accuracy, i.e., the number of pairs where the human correctly guessed which tweet will be retweeted. The subjects were asked to do two experiments: in the first one they were only presented the text of the tweets and asked to make a decision, and in the second one (carried out few days after the first one), they were also presented all the social information described in Section 4.3.

The pairs of tweets were chosen randomly from the test set, under the following constraints: i) both tweets had to be written within one hour of each other, ii) they had to be in English (enforced by author's choice of language on his profile), and iii) authors of both tweets had to be from the same time zone as the human subjects. We do this in order to remove the biases due to i) time of day when the tweet was written (our prior analysis showed that tweets written in different times of day have a different probability of being retweeted), ii) language – human subjects only spoke English so presenting one tweet in English and one in Japanese would lead to an obvious bias, and iii) difference in geographical locations. If we presented the human subjects with a tweet written by an author from, say, India, and a tweet written by an author from their own time zone, it is very likely that the subject could not properly asses the likelihood of the first tweet being retweeted.

In the first experiment, both subjects significantly beat the random baseline (which gets a 50% accuracy): the first subject had an accuracy of 76.2%, and the other 73.8%. This shows that humans are indeed capable of distinguishing tweets which will get retweeted from those which will not, and based on the content of the tweet alone. In the second experiment, when all the social information was shown too, the first subject got an accuracy of 81.2%, and the second subject 80.2%. Thus, the human subjects were able to make use of this information to improve their prediction, indicating that there is useful information in the social features.

5.2 Results

Predicting retweets is a binary classification task – each tweet is assigned a label 0 (will not be retweeted) or 1 (will be retweeted). Because we are more interested in correctly predicting when something is retweeted than correctly predicting when something is not retweeted, we use the F_1 score to measure performance of our models, as is standard in information retrieval where there is a similar imbalance between the relevant and non-relevant classes.

Table 1 shows performance of different classifiers on the task of predicting what will be retweeted. The majority classifier in third column simply predicts that everything will be retweeted.[2] Both PA and time-sensitive models used all the features introduced in Section 4.3, and the feature values were normalized to be in the $[0, 1]$ interval. We can see that just using PA already gives us a huge improvement in F_1 over both baselines. Additionally, using local models for different times a tweet was written in together with a global model (such as the one in column four) yields another 9 points improvement in F_1. This suggests that more work should be done in taking into account different retweet patterns across time.

Time-sensitive approach. We are interested to find how much does our time-sensitive approach contribute, and which features benefit most from it. Table 2 shows that social features alone perform very well, which means that a lot of retweets are about who we are. However, if we recall that the best model achieved an F_1 score of 46.6, we can see that there is definite benefit in looking at what is actually written in the text of the tweet. Looking at the tweet features only, we see that they outperform both baselines from Table 1, but do slightly worse than social features. Hence, it is possible to predict retweets just by looking at the text of the tweet, although it is better to know who wrote it. We can see that using a threshold to remove infrequent words actually hurts performance, and we thus use all the words. Note, however, that the decrease in performance is not big if a threshold of 10 is used, and it thus might still be beneficial to use it in order to reduce memory usage and speed up computation (using a threshold of 10 reduces the number of features from $\sim$ 24 million to $\sim$ 700 thousand).

Finally, we look at how different feature groups benefit from our time-sensitive approach. We can see that tweet features show a substantial improvement of about 10 points in F_1 score when using this model. On the other hand, social features seem to benefit less from this approach which suggests that words have very different effect on retweet probability depending on the creation time of the tweet, whereas social features show less variation with respect to tweet's creation time.

Important social features. To further understand how individual features contribute, we look at the weights our model assigns to them. Features with the highest weights in the global component of the best performing model are *listed* (0.48), *# of followers* (0.46), and *# of friends* (0.19). We can see that the number of times a user was listed

[2] A majority classifier which predicts that nothing will be retweeted simply gets an F_1 score 0, as there are no true positives.

Model	Random	Majority	PA	Time-sensitive
F_1	11.9	12.7	37.6	**46.6**

Table 1: Comparison of different models on the retweet prediction task. The C parameter for PA models was set to 0.01. Higher is better.

Model	F_1	F_1 (TS)
Social only	38.5	39.6
Tweet only (full)	22.4	35.0
Tweet only (t = 10)	22.4	34.1
Tweet only (t = 100)	20.6	29.7

Table 2: Performance of each group of features, and the benefits from using a time-sensitive model. F_1 (TS) indicates the performance of the features when time-sensitive modeling is used. Parameter t for lexical models denotes the cutoff threshold.

plays a very important role, along with the number of followers. The importance of number of followers for predicting retweets has been noted before (Suh et al. 2010; Zaman et al. 2010), but we are not aware of any other research that has examined the importance of Twitter's lists, and this result suggests that more work should focus on the lists. The most negative weights were assigned to *# of mentions* (-0.14), and *# of statuses* (-0.11).

What are the "good" and "bad" tags? We further analyzed which hashtags received high positive or negative weights in our best model. Top positive hashtags were #ff (follow friday), #ww (woof Wednesday), #fblikes, and various teenage-related topics (#ohmyteenager, #omgteenquotez, #dailyteen, #teenagersfacts, ...), whereas the top negative weighted topics included #newfollower, #follow4follow, #followyouback, #instantfollow, and the like. This seems to show that asking for followers will make one's tweet less likely to be retweeted (except on a Friday), whereas writing about teen-related things will increase the chances of being retweeted.

Comparison with human results. Finally, to put these numbers in perspective, we turn back to the human experiment from Section 5.1. We use our models to predict which of the two tweets presented is more likely to be retweeted, much in the same way the humans were asked to do. To compare against human performance in the first task (when the humans were presented only with tweets), we train a model using only tweet features to make sure it does not use any information that the humans did not. To make the comparison with human subjects fair, we train our model on English tweets only. This model achieved an accuracy of 69.3%, which is lower than human subjects, but not significantly different from either subject at $p = 0.05$. In the second experiment, we compare our best model (last column in Table 1) to the human performance when they were presented with all the social information in addition to the text of tweet. Our model achieves an accuracy of 82.7%, slightly higher than both human subjects, but not significantly different at $p = 0.05$. This is a very encouraging result which shows that for the task of predicting retweets we are able to devise algorithms that do as well as humans.

6 Conclusion

A fundamental task to understanding retweeting is predicting whether a tweet will be retweeted, which is the focus of this paper. Before trying to build models to predict retweets, we conducted a human experiment which showed that humans are capable of doing this task, meaning that the task is indeed possible. We used a machine learning approach based on the passive-aggressive algorithm and showed that it substantially outperforms the baseline. We proposed a time-sensitive model that builds separate models depending on the tweet's creation time and showed that this substantially improves performance over the vanilla PA algorithm. Analyzing the features, we found that social features perform very well, but the model does benefit from using tweet features. A comparison of our model with human subjects showed that our model performs as well as humans.

References

Boyd, D.; Golder, S.; and Lotan, G. 2010. Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. In *Proceedings of the 43rd Hawaii International Conference on System Sciences*, volume 0, 1–10. Los Alamitos, CA, USA: IEEE Computer Society.

Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; and Singer, Y. 2006. Online passive-aggressive algorithms. *The Journal of Machine Learning Research* 7:551–585.

Nagarajan, M.; Purohit, H.; and Sheth, A. 2010. A Qualitative Examination of Topical Tweet and Retweet Practices. In *Proceedings of Fourth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 295–298.

Suh, B.; Hong, L.; Pirolli, P.; and Chi, E. H. 2010. Want to be retweeted? Large scale analytics on factors impacting retweet in twitter network. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, 177–184. IEEE.

Yang, J., and Counts, S. 2010. Predicting the speed, scale, and range of information diffusion in twitter. In *International AAAI Conference on Weblogs and Social Media*, 355–358.

Zaman, T. R.; Herbrich, R.; van Gael, J.; and Stern, D. 2010. Predicting information spreading in twitter. In *Workshop on Computational Social Science and the Wisdom of Crowds , NIPS*.