On Predicting Deletions of Microblog Posts

Mossaab Bagdouri
Department of Computer Science
University of Maryland
College Park, MD, USA
mossaab@umd.edu

Douglas W. Oard
iSchool and UMIACS
University of Maryland
College Park, MD, USA
oard@umd.edu

ABSTRACT

Among the many classification tasks on Twitter content, predicting whether a tweet will be deleted has to date received relatively little attention. Deletions occur for a variety of reasons, which can make the classification task challenging. Moreover, deletion prediction might serve different goals, the characteristics of which should be reflected in the evaluation design. This paper addresses the problem of deletion prediction by analyzing the distribution of deleted tweets, presenting a new evaluation framework, exploring tweet-based and user-based features, and reporting prediction scores.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Measurement, Performance

Keywords: Microblogs; Prediction; Deletion

1. INTRODUCTION

Social media platforms have garnered substantial attention from researchers in recent years. The abundance of public content that is available from some services, perhaps most notably Twitter, has inspired a wide range of applications such as characterizing user demographics [7], predicting future events [2] and explaining sociopolitical interactions [10]. Among this work, the thread closest to our focus has explored predicting the reaction of users to specific microblog posts (in Twitter, "tweets"). Work in this thread has included prediction of replies [8], retweeting [5], and deletion[6]. Of these, prediction of deletion has received the least attention to date. A public message on a microblogging service might be deleted for several reasons and in different ways. For instance, some content may be removed by a third party due to a perceived violation of some law, regulation, or norm [1]. In other cases, a person might regret posting an embarrassing comment, later choosing to delete their own tweet [9]. Alternatively, a user might become concerned about privacy and opt to make their profile private, which in Twitter results in deletion of all of their public tweets (these tweets may still be accessible to a restricted number of accounts; they will show as deleted for everybody else). A similar effect can be achieved by Twitter itself, for example when an account is suspended because of spamming or posting inappropriate content. Adding to the complexity, deleting a tweet automatically generates a cascade of deletions for all retweets of that tweet.[1]

Predicting whether a tweet will be deleted might have several applications. Perhaps most obviously, we might help a user avoid posting comments that they may later regret. We might also alert users before they post content that is similar to what has previously been censored. Of greatest importance in our own work, if we can predict which tweets are likely to be deleted, we can act proactively to minimize the "swiss cheese" decay of Twitter test collections in which we have invested annotation effort on tweets that will later be deleted, and thus unusable (by Twitter's terms of service) by future users, including our future selves.

This paper addresses the task of predicting whether a tweet will be deleted. Although we are interested in deletions over many time scales, in this paper we focus on predicting which Tweets will be deleted within 24 hours of being posted. After introducing the datasets and our classifier design in Section 2, we use a simple example to illustrate the importance of reflecting the task characteristics in the evaluation design (Section 3.1). We then propose two new evaluation designs in Sections 3.2 and 3.3, reporting classification results for each. Finally, we summarize our findings and make some remarks on future work in Section 4.

2. METHODS

We present two datasets that we collected to explore the problem of deletion prediction and we describe our classifier design, with particular attention to feature selection.

2.1 Datasets

To conduct our experiments, we collected two datasets in a similar way. We use the first dataset, DS1, to explore the problem and the second, DS2, to validate our results. In our research we are particularly interested in the use of Twitter for Arabic, but the Twitter API does not support language selection alone without specifying some content terms. We therefore started with a list of the most frequent 400 terms

[1]A retweet is a reposting of someone else's tweet.

CIKM'15, October 19–23, 2015, Melbourne, VIC, Australia.

ACM 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806600 .

Table 1: Summary of the characteristics of the datasets DS1 and DS2.

	DS1	DS2
Seed time	10/24/14 13-14	12/19/14 14-15
Streaming started	10/24/14 14:53	12/21/14 16:58
Streaming ended	11/21/14 01:21	01/22/15 23:59
Users followed	95,000 users	180,000 users
Users who tweeted	91,283	179,425
Number of tweets	80,8239,916	415,582,993
Labeled tweets	78,527,525	406,140,249
Deletion rate	3.64%	2.33%
Mean del rate by user	3.55%	2.88%
S.D. del rate by user	9.15%	7.47%

in a set of 1,050,000 tweets streamed from this API for which Twitter had classified the tweet as Arabic. We then use the API to obtain tweets that contain any of these terms, with a restriction to tweets classified by Twitter as Arabic, for one hour. Using these tweets, we then uniformly sampled a substantial subset of unique users out of those who sent at least one tweet during that hour and we then used the Twitter API to "follow" those users for about a month, tracking both their tweets and any subsequent deletion notifications. Some of the users did not tweet again during the month; we exclude them from the dataset. After sorting all of the tweets chronologically, we used a sliding window of 24 hours to detect which tweets were deleted within one day of being posted. The last tweet we consider is the one received 24 hours before the end of our collection; this allows us to study deletion over a time period of the same duration (one day) regardless of when the tweet arrived. We then labeled each tweet as deleted or not. To comply with Twitter's terms of service, we built our features as soon as the tweet arrived; for deleted tweets we retain the features in order to conduct our study, but we do not retain a readable form of the tweet. Table 1 summarizes the statistics of these datasets.

2.2 Classification

The deletion prediction task exhibits a strong class imbalance, with many fewer positive than negative instances. We prefer an evaluation measure that emphasizes correct decisions on the minority class (deletion), which is the class of interest in this case. We have therefore chosen F_1, the balanced harmonic mean of recall and precision, as our measure of effectiveness. Accuracy is also a commonly reported measure, but on this task accuracy results would be dominated by results on the majority class (not deleted). We would prefer to optimize the classifier directly for F_1 (e.g., using SVM-perf), but this is currently only practical for datasets that are small enough to fit in memory [4]. Given the scale of our datasets, the efficient online classifier Vowpal Wabbit [11] is the better choice. We therefore configured Vowpal Wabbit with logistic regression as the loss function, using default settings for other parameters. We split each dataset into three subsets of 70%, 10% and 20% corresponding respectively to training, development and testing. Vowpal Wabbit assigns a deletion prediction score to every tweet. We use a grid search to find the threshold on that score that maximizes F_1 on the development set. We then classify the testing set using that threshold.

2.3 Feature Selection

Feature selection can have a substantial impact on classifier accuracy. We therefore implement a recursive feature elimination algorithm [3] using DS1. We start with all the features that are available as fields in the JSON object retrieved from Twitter API, which are either tweet-based or user-based. We then derive some synthetic features, such as the word count in the content of a tweet, and the hour of the day when the tweet was created. The features that are provided as a single value (e.g., language of tweet, total number of tweets published by a user) are removed one at a time, whereas those that are provided as a list of values (e.g., bag of words, list of URLs) are removed together at once. Each feature that improves the F_1 value on the DS1 test set when removed during this ablation process is excluded from the feature set. We repeat this process until we settle on a final set of features that each deteriorates the F_1 score on the DS1 test set when (individually) removed. We perform the feature selection process separately for each evaluation condition that we study, but we report DS2 results using the features selected for the same condition using DS1. Our reported DS2 results are thus "fair," in the sense that we believe them to be representative of a classifier that we could actually run on unseen data.

3. PREDICTING DELETIONS

We start the prediction task by observing that a feature as simple as the user ID predicts deletions better than any combination of available features, given the evaluation setup of previous work. We then suggest two alternative evaluation settings and report the corresponding features and scores.

3.1 Naive Features and Evaluation

Petrovic et al. report that the best performance they could achieve was based on training a Support Vector Machine using social and text features in addition to the user IDs [6]. Using their features on our (different) data, we achieve a score of 0.387 on dataset DS1. Surprisingly, however, our feature selection ablation process achieves an even better result with just a single feature, obtaining an F_1 of 0.455 on DS1 just by using the user IDs. Interestingly, Petrovic et al. saw just the opposite, reporting that F_1 fell from 0.270 (for their full feature set), to 0.122 (for user ID alone).

This finding, that on our data (but apparently not on Petrovic et al.'s data) a classifier that learns that some users will often delete their tweets while others will rarely do so does well, suggests a very sharp skew in the data. This led us to investigate the distribution of deletions across all of the users. We do so by first plotting, on a log-log scale, the number of deleted tweets for each user, sorted in a decreasing order, in Figure 1. The resulting plot is piecewise linear, indicating spliced power law distributions that result in a fat head (i.e., an unexpectedly large number of prolific deleters) and thus a thinner tail. The net effect of this is easily seen in the cumulative sum of deletions in Figure 2; this is the area under the (linear scale) deletions-vs.-rank plot. We observe that the first percentile of 912 users is responsible for about a half of all deletions (1,352,043 deletions, 47%), and that 10% of the users (9,127) are responsible for more than four out of every five deletions (i.e., 2,331,268 deletions, 81%). Examining a few new tweets from some of these most prolific deleters suggests to us that many of these prolific deleters may be automated systems that are engaged in advertising

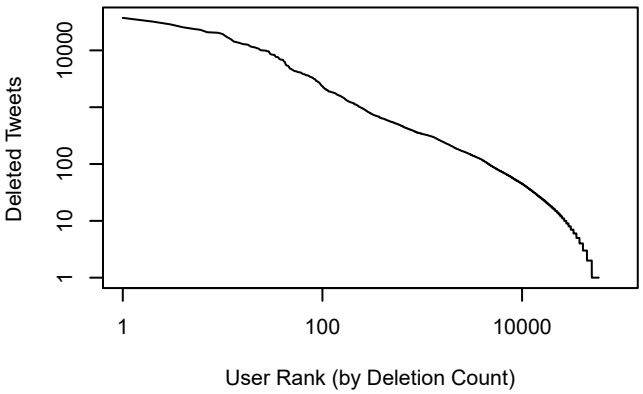

Figure 1: The number of deletions per user for the dataset DS1 in a descendant order. Both axes are in log scale. The deletions appear to follow spliced power-law distributions.

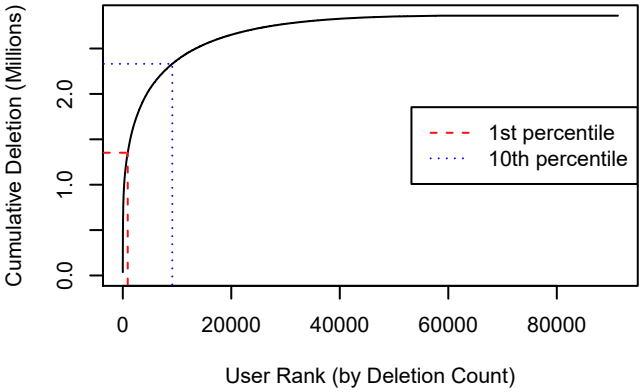

Figure 2: The cumulative count of deletions for the dataset DS1, with a descending sort of users based on their deletes.

activities. Depending on the deletion task, we may, or may not, want to exclude them from our training and prediction. Indeed, for a task such as cleaning a dataset of spam, we would actually want such accounts in the dataset. For a different task such as predicting regret, we might want to focus more on helping real (i.e., human) users and thus we would prefer not to train or test on accounts like these that distort the actual deletion patterns that we wish to learn.

Another aspect of the distribution of the deletions is related to the presence of retweets, since if an original tweet is deleted, then all of its retweets are also automatically deleted.[2] Again, depending on the specifics of the deletion task, we may or may not want to include retweets in the training, prediction and evaluation. For example, we may want to predict that a retweet will be deleted (perhaps because the retweeting user follows several spammers) in a dataset cleaning task. For regret, we may want to restrict our focus to original tweets.

3.2 Separating Users

For this experiment, we want to neutralize the effect of the user ID in the most direct way possible. Simply excluding the user ID from the feature set would not suffice, since a combination of other features (e.g., name, description, number of tweets) might serve as an effective surrogate for the user ID. Even if we were to ignore all user-based features, tweet content might even serve as an effective proxy for the identity of some users (e.g., for users who repeatedly send some distinctive message). For this reason, we opt to split our training, development and test subsets randomly by users rather by tweets, maintaining the respective 70/10/20 ratios. We then use the procedure described in Section 2.3 to select the best features on dataset DS1, which we show in Figure 3. In this plot, the best performance (0.298) is achieved when we include all of the tweet (T) and user (U) features. Excluding all of the features corresponds to predicting all the tweets in the test set to be deleted, which gives a baseline F_1 score of 0.064. An individual or a set of features is either excluded—showing its contribution to the whole combination—while all of the other features are included, or is included—to show its individual contribution compared to the baseline—while the other features are excluded. In combination, the tweet features are stronger predictors than the user features, with the tweet content having the greatest predictive power, both individually, and in combination with other features. The source of the tweet—a field indicating how the tweet was published (e.g., through a smartphone, using a third-party application, or from a desktop browser)—is the second strongest tweet-based feature (by the "Excluded" plot). This is consistent with Sleeper et al.'s qualitative survey in which they report that 45% of regretted tweets were made from a mobile device [9], suggesting a signal might exist from knowing the type of device used. The user-based features add to what can be achieved using the tweet-based features alone, increasing the F_1 score from 0.261 to 0.298.

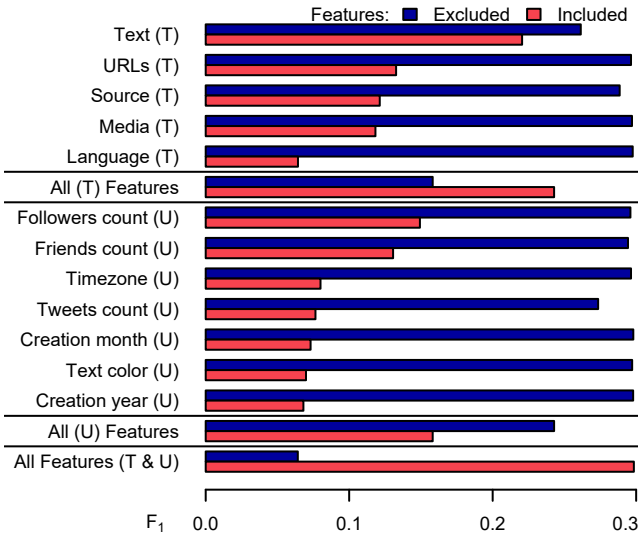

Figure 3: Best features for the dataset DS1 when we split by users. The best performance is achieved when all of the tweet-based (T) and user-based (U) features are included.

We apply these features to dataset DS2 to obtain an F_1 score of 0.375. We also applied Petrovic et al.'s features on this dataset, getting an F_1 score of 0.356. This suggests that on similar unseen data collected as described in Section 2.1 and split by users, we would expect the features we found somewhat to perform better than those of Petrovic et al.

[2] https://support.twitter.com/entries/18906

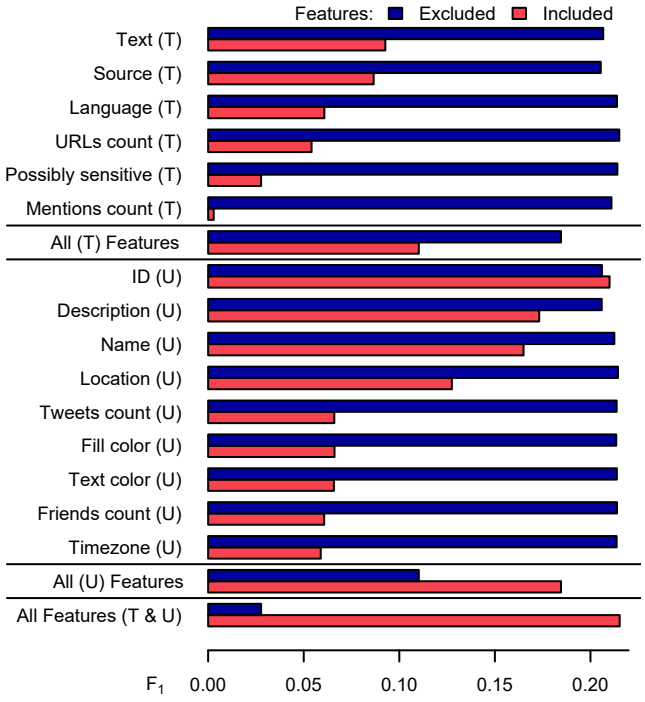

Figure 4: Best features of the dataset DS1 for the chronological split, after excluding the retweets and outliers.

3.3 Excluding Outliers

We now want to maintain the effect of the user information but diminish the noise generated by outlier users. To do so, we first eliminate all of the retweets. This causes the ratio of deleted tweets to go down from 2,861,663 / 78,527,525 = 3.6% to 1,007,760 / 46,233,510 = 2.2%. In other words, the deleted retweets represent 65% of all of the deletions. We are left with 87,469 users (out of 91,274, i.e., 96%) who have posted at least one tweet that is not a retweet. Next we exclude the 2,049 users who have a deletion rate larger than the average deletion rate of these 87,469 users by three standard deviations, because we consider them to be outliers. In fact, these 2% of users are responsible for the deletion of 340,136 tweets that are not retweets, or just above one third of all deletions that are not of retweets. Finally, we have a set of 45,611,445 tweets that are neither retweets nor posted by outliers, containing 667,624 deletions.

Figure 4 shows the best features as found by the process described in Section 2.3. We see that the user ID is still the dominant feature, as it accounts for 0.210 out of the total F_1 score of 0.215. Other user-based identifiers such as the description and the name are good substitutes. Incorporating other user-based features (without any tweet-based features) does actually hurts, reducing the F_1 score to 0.185. Adding tweet-based features slightly improves the F_1 score to 0.215.

We apply these features to dataset DS2 to obtain an F_1 score of 0.188, about the 0.185 that we get with Petrovic et al.'s features. Because user ID is present in both feature sets, we can conclude that removing outliers in the way we have done does little to affect the relative value of the user ID feature (although it does drop the best achievable F_1 by quite a lot, from 0.455 to 0.210, suggesting that user ID is even more useful for the outlier users than for others).

4. CONCLUSION AND FUTURE WORK

We addressed the task of tweet deletion prediction. Among a set of features directly available for a tweet, we found that the user ID is a surprisingly strong feature for both of our datasets, even after removing outliers. From the fact that Petrovic et al. did not see this, we can conclude that our Arabic Twitter datasets have somewhat different characteristics from theirs. Depending on the specifics of the task, we have identified some characteristics that should be considered in the design of evaluation methods, such as the inclusion or exclusion of retweets and the inclusion or exclusion of users with abnormal deletion activity. We have suggested some appropriate feature sets, and reported performance scores for such cases. While we allowed a period of 24 hours to detect deletions in our experiments, we do not know if various deletion types (e.g., regret deletions, cascade of deletions of retweets, spam deletions, etc.) take place at similar or different times. Conducting such a study would require us to build a classifier that differentiates between various types of deletions. In future work, we are also interested in studying the effect of linguistic features on the deletion prediction. Indeed, it is not clear yet whether the behaviors we observed in a dataset where Arabic is the dominant language and the Arab world is the likely originating location will be applicable to datasets of other languages and regions.

ACKNOWLEDGMENT

This work was made possible by NPRP grant# NPRP 6-1377-1-257 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. We thank Maram Hasanain for helping us select the 400 words frequent in Arabic tweets.

5. REFERENCES

[1] D. Bamman et al. Censorship and deletion practices in Chinese social media. *First Monday*, 17(3), 2012.
[2] J. Bollen et al. Twitter mood predicts the stock market. *J. of Comp. Science*, 2(1):1–8, 2011.
[3] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.
[4] T. Joachims and C.-N. J. Yu. Sparse kernel SVMs via cutting-plane training. In *ECML PKDD'09*.
[5] S. Petrovic, M. Osborne, and V. Lavrenko. RT to win! Predicting message propagation in Twitter. In *ICWSM'11*.
[6] S. Petrovic, M. Osborne, and V. Lavrenko. I wish I didn't say that! Analyzing and predicting deleted messages in Twitter. *CoRR*, abs/1305.3107, 2013.
[7] D. Rao et al. Classifying latent user attributes in Twitter. In *SMUC'10*, pages 37–44.
[8] M. Rowe, S. Angeletou, and H. Alani. Predicting discussions on the social semantic web. In *ESWC'11*.
[9] M. Sleeper et al. "I read my Twitter the next morning and was astonished:" A conversational perspective on Twitter regrets. In *CHI'13*, pages 3277–3286.
[10] K. Starbird and L. Palen. (How) will the revolution be retweeted?: Information diffusion and the 2011 Egyptian uprising. In *CSCW'12*, pages 7–16.
[11] K. Weinberger et al. Feature hashing for large scale multitask learning. In *ICML'09*, pages 1113–1120.