A Personalized Global Filter To Predict Retweets

Michail Vougioukas
European Organization for Nuclear Research

Ion Androutsopoulos
Athens University of Economics and Business

Georgios Paliouras
National Center for Scientific Research ”Demokritos”

KEYWORDS

Twitter, personalization, user modeling, machine learning.

Introduction. Information shared on Twitter is ever increasing and users-recipients are overwhelmed by the number of tweets they receive, many of which of no interest. Filters that estimate the interest of each incoming post can alleviate this problem, for example by allowing users to sort incoming posts by predicted interest (e.g., 'top stories' vs. 'most recent' in Facebook).

Global [1, 5] and *personal* filters [2] have been used to detect interesting posts in social networks. Global filters are trained on large collections of posts and reactions to posts (e.g., retweets), aiming to predict how interesting a post is for a broad audience. In contrast, personal filters are trained on posts received by a particular user and the reactions of the particular user. Personal filters can provide recommendations tailored to a particular user's interests, which may not coincide with the interests of the majority of users that global filters are trained to predict. On the other hand, global filters are typically trained on much larger datasets compared to personal filters. Hence, global filters may work better in practice, especially with new users, for which personal filters may have very few training instances ('cold start' problem).

Method. Following Uysal and Croft [2011], we devised a hybrid approach that combines the strengths of both global and personal filters. As in global filters, we train a *single* system on a large, multi-user collection of tweets. Each tweet, however, is represented as a feature vector with a number of *user-specific features* (Fig. 1).

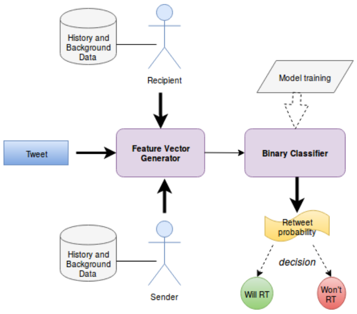

Figure 1: Architecture of our system.

A tweet received by two users is represented by two different feature vectors. This allows the system to consider user preferences and produce different predictions per recipient, as in personal filters, while still being able to generalize over different users.

Our system predicts how likely it is that a particular user (the *recipient* of Fig. 1) will retweet a particular incoming tweet. The system has access to the history of tweets of the recipient and of the sender, as well as background information about the recipient and the sender, which we obtained through the Twitter API (http://dev.twitter.com/rest/public). By *sender* we mean the user that caused the recipient to receive the tweet, either by authoring it directly or by retweeting it. The tweet is represented as a recipient-sensitive feature vector, which is passed onto a logistic regression classifier that predicts if the recipient will retweet the incoming tweet or not. The classifier is trained on tweets received by Twitter users and the users' decision to retweet the incoming tweets or not. Using previous retweet actions as gold labels for training has the advantage that no extra human labeling of tweets with interest scores is required to construct the datasets.

The feature vector of each incoming tweet contains up to 50 features, describing factors possibly affecting a tweet's probability to be retweeted. The feature set was mostly inspired by past work and contains 7 groups of features. Group 1 contains features that examine the tweet itself (e.g., tweet length). Group 2 contains features that examine how similar the incoming tweet is to particular collections of tweets (e.g., all sender's previous posts). Group 3 contains features modeling the network influence and authority of the sender and the recipient (e.g., account statistics from Twitter and Klout.com). Group 4 contains features that capture the interaction between the sender and the recipient. Group 5 contains features that estimate the timeliness of the incoming tweet by measuring its similarity with, e.g., other recently received tweets. Group 6 contains features related to the users the recipient follows. Group 7 complements Group 1 by looking for special keywords and parts of speech in the tweet (using the CMU ARK Twitter tagger [3]).

Experiments. We experimented with 122 English-writing journalists, as recipients. We used their retweets as positive instances and a random subset of posts of users they follow as negative instances. We merged all journalists' data into 140 temporally ordered, balanced (equal number of positive and negative instances) *batches*. The first 120 batches were used as the *training set*, the next 10 batches were used as the *balanced development set*, and the last 10 batches were used as the *balanced test set*. The balanced data sets are not realistic, because they assume that receivers retweet on average half of their incoming tweets. Thus, we also constructed *unbalanced development and test sets* by randomly downsampling the positive instances of each batch, leaving 5% positive and 95% negative instances (a ratio which, based on estimations on our dataset, is realistic). We trained our logistic regression classifier on the balanced training set and we evaluated it on both the balanced and the unbalanced development and test sets. As expected,

UMAP'17, July 9–12, 2017, Bratislava, Slovakia

ACM ISBN 978-1-4503-4635-1/17/07.
DOI: http://dx.doi.org/10.1145/3079628.3079655

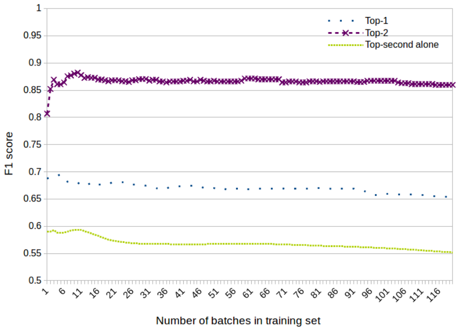

Figure 2: F1 on the unbalanced development set, using only the top feature, only the 2nd-top, or both.

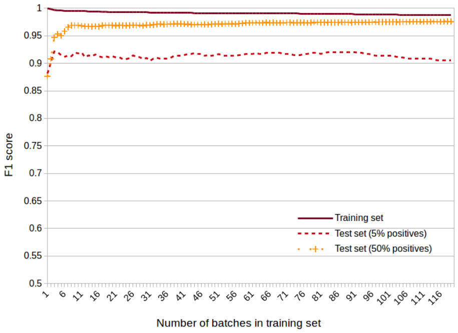

Figure 4: F1 on the balanced and unbalanced test set vs. F1 on the (always balanced) training set, using the top 10 features.

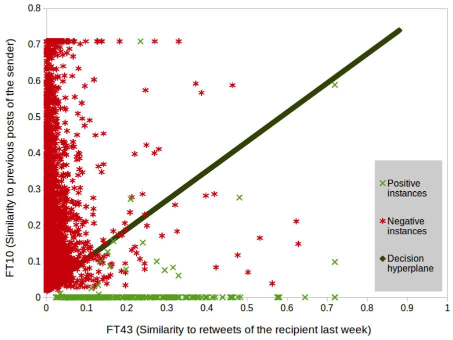

Figure 3: Sample positive and negative instances from the unbalanced development set and the linear separator learned.

the balanced development and test sets proved to be easier for the classifier than their unbalanced counter-parts (Fig. 4).

To estimate the usefulness of each feature, we ranked them by decreasing correlation to the class label. We, then, evaluated the system with the *F1 score*, when using the top-k features ($1 \leq k \leq 50$). We evaluated it in an incremental, w.r.t. the size of the training set, manner; the experiment was repeated 120 times, each time training the classifier on the earliest m batches of the training set ($1 \leq m \leq 120$), always using the same development set to evaluate the performance of the classifier for each value of m.

We witnessed a notable change in the F1 score when the second top feature (FT10: *similarity to tweets previously posted by the sender*) was added to the top one (FT43: *similarity to tweets retweeted by the recipient during the previous week*). Figure 2 shows the F1 score, using only the top feature, only the second-top, or both. The second-top feature alone is not a good predictor, but the combination of the two features increases F1 to ≈ 0.87, which is close to the best score that we obtained ($F_1 \approx 0.9$ for the top 10 features).

Figure 3 sheds more light on the role of the top two features. The straight line is the separator the classifier learned. In most cases, it correctly separates the negative (stars) from the positive (crosses) instances, which agrees with the high F1 score in Figure 2. As expected, most negative instances have low similarity to the tweets the recipient retweeted during the previous week. Thus, the recent retweets of the recipients are good indicators of their current interests. Unexpectedly though, most positive examples have very *low* similarity to the previous posts of the sender. Recipients tend to prefer posts that are *unusual* for the particular sender. We, finally, observe that negative instances tend to have small FT43 values, but a non-negligible subset of positive instances also has small FT10 values. Most of those positive instances, however, have near-zero FT10 values, unlike most negative instances and, hence, the combination of the two top features boosts the classifier.

Finally, we evaluated our system on previously unseen examples, using both the balanced and the unbalanced test sets. As expected, the system was found to perform better on the balanced test set and worse on the unbalanced test set. The gap between the system's performance on the training and balanced test data is small, indicating that the system does not significantly overfit the training data. Overall, when trained on a dataset of approx. 130K tweets received by 122 journalists, our system obtains $F_1 \approx 0.9$ using only 10 features and only approx. 5K training instances (Figure 4).

Conclusions. The main contributions of this paper are: (a) a lightweight retweet prediction model, which attains high F1 score with few features and training instances; (b) a thorough investigation of most features mentioned in related literature and variants thereof, grouped into feature types for further research; (c) a large dataset of tweets and related user information, which we plan to make available in an encoded form. A longer version of this paper can be found at http://nlp.cs.aueb.gr.

REFERENCES

[1] O. Alonso, C. C. Marshall, and M. Najork. 2013. Are some tweets more interesting than others? #hardquestion. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval.* Vancouver, Canada, 2.

[2] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. 2010. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the Conference on Human Factors in Computing Systems.* Atlanta, USA, 1185–1194.

[3] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. Smith. 2011. Part-of-Speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies.* Portland, USA, 42–47.

[4] I. Uysal and W. Bruce Croft. 2011. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the International Conference on Information and Knowledge Management.* Glasgow, Scotland, 2261–2264.

[5] M. Yang and H. Rim. 2014. Identifying interesting Twitter contents using topical analysis. *Expert Systems with Applications* 41, 9 (2014), 4330–4336.