Mining Twitter Messages for Software Evolution

Guzman Ortega, Emitza ; Ibrahim, Mohamed ; Glinz, Martin

DOI: https://doi.org/10.1109/ICSE-C.2017.65

2017 IEEE/ACM 39th IEEE International Conference on Software Engineering Companion

Mining Twitter Messages for Software Evolution

Emitza Guzman
University of Zurich
guzman@ifi.uzh.ch

Mohamed Ibrahim
Technische Universität München
m.ibrahim@tum.de

Martin Glinz
University of Zurich
glinz@ifi.uzh.ch

***Abstract*—Twitter is a widely used social network. Previous research showed that users engage in Twitter to communicate about software applications via short messages, referred to as tweets, and that some of these tweets are relevant for software evolution. However, a manual analysis is impractical due to the large number of tweets – in the range of thousands per day for popular apps.**

In this work we present ALERTme, an approach to automatically classify, group and rank tweets about software applications. We apply machine learning techniques for automatically classifying tweets requesting improvements, topic modeling for grouping semantically related tweets and a weighted function for ranking tweets according to their relevance for software evolution.

We ran our approach on 68,108 tweets from three different software applications and compared the results against practitioners' assessments. Our results are promising and could help incorporate short, informal user feedback with social components into the software evolution process.

***Index Terms*—user feedback, software evolution, text mining.**

I. Introduction

Twitter receives over 500 million short messages, tweets, per day and is one of the most popular social networks. In our previous work [1] we found that users communicate about software applications through Twitter and that some of these tweets contain information that can be relevant to software evolution such as bug reports, feature requests and feature shortcoming descriptions. As in the case of user reviews from app stores, tweets could embody the users' voice and be used to drive the software evolution effort.

However, due to the large number of tweets about software applications a manual analysis is infeasible [1]. Recent research has focused on the mining of user feedback from app stores for software evolution purposes, e.g., [2], [3], [4]. While tweets and app store reviews share similarities, such as their high numbers, unstructured nature and informal language, they also have significant differences in respect to their length and available metadata. Therefore, it is necessary to investigate the extent in which data mining techniques that have been previously used for the automatic analysis of app reviews can be used on tweets about software applications.

For this purpose, we present ALERTme (A LittlE biRd Told me), an approach to classify, group and rank tweets for their use during software evolution.

II. The ALERTme approach

The overall goal of ALERTme is to automatically classify, group and rank tweets that are relevant for software evolution tasks. Figure 1 shows its main steps. First, we *preprocess* the tweet text using natural language techniques. Second, we *classify* the text present in each tweet into two categories using supervised machine learning. This step allows for the removal of irrelevant information. Then, we *group* the tweets that are relevant for software evolution by using a topic modeling algorithm specialized in short text. After the execution of this step, we obtain groups of tweets with semantically similar content. These groups can be used as a tweet summary. Last, we *rank* the tweets by using a weighted function on several tweet attributes, such as number of duplicates and tweet sentiment. In the following sections we describe each of the approach steps in additional detail.

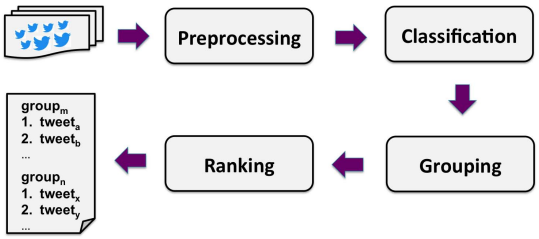

Fig. 1. ALERTme overview.

A. Preprocessing

We prepare the input data by performing the following steps on the tweet text: (1) tokenization, (2) lower case conversion, (3) stopword removal and (4) stemming.

B. Classification

The goal of this step is to automatically classify the tweets into two categories: *improvement request* and *other*. We define an *improvement request* as all tweets that call for enhancements to the application i.e., bug reports, feature requests and feature shortcoming descriptions. All tweets that do not fall into this category are considered in the *other* category.

We use Multinomial Naive Bayes for the task. This decision is motivated by the good performance of Naive Bayes classifiers when classifying text [5] and its promising results when solving other software engineering tasks.

C. Grouping

We use a topic modeling algorithm specialized in short text, Biterm Topic Model (BTM) [6], for grouping semantically related tweets. BTM takes as input the preprocessed text of each tweet and outputs the *topics*, i.e., groups of words that

978-1-5386-1589-8/17 $31.00 © 2017 IEEE
DOI 10.1109/ICSE-C.2017.65

co-occur in the whole corpus of tweets. The set of words {*crash, update, frustrated, new, version, bug*} is an example of a topic related to a users' experience when updating a software application.

D. Ranking

We rank tweets according to their relevance, i.e., how fast those involved in software evolution (e.g., developers or project managers) should react to the tweet. For this purpose we use the following weighted function R, ranking tweet tw as:

$$R(tw) = \sum_{k=1}^{6} w_k * c_k(tw) \quad (1)$$

where c_k are the ranking coefficients for specific tweet attributes k and w_k are the attributes' manually assigned weights. In our work we consider six specific tweet attributes:

- *Retweets*: a retweet is the republishing of a tweet. The retweet number of a particular tweet allows to estimate its reach. A tweet with a high retweet count will reach many people and might suggest that a high proportion of users are reporting the same issue.
- *Likes*: likes are indicators of appreciation towards the concerned tweet. The number of likes could be an indicator of the amount of people who find the tweet interesting or are facing the same issue.
- *Social Rank*: in Twitter, users can follow other users, friends, or have users following them, followers. The number of followers and friends can be important factors when predicting the influence of a Twitter user [7]. We define social rank as the number of followers multiplied by the ratio of followers to friends. This computation allows to circumvent tweets from bots and users with aggressive following behaviour.
- *Content category*: the category of the tweet in regard to its content and software evolution. Tweets leading to improvements in the software application could be more relevant. We use the results from the classification step of our approach to obtain this information.
- *Duplicates*: the number of tweets that are lexically or semantically similar to a specific tweet. Duplicate tweets could indicate that several users are discussing the same issue. In our current work we use Jaccard similarity for the measurement of lexical similarity. The results from the grouping step of ALERTme could also be used as an indicator for semantic similarity.
- *Sentiment*: sentiment is the affect or mood expressed in a tweet. For example, a tweet can have a very positive, neutral or very negative sentiment. Tweets displaying a high negative sentiment, could indicate a high user dissatisfaction and might indicate a need for special attention. We use SentiStrength [8] a lexical sentiment analysis tool specialized in short, informal text for the extraction of sentiments.

A description of the coefficient computation of each attribute and its respective weight assignment is available in our oncoming work [9].

III. Evaluation

We ran ALERTme on 68,108 tweets about three software applications: Dropbox, Slack and Spotify. Specifically, we trained and evaluated our classifier with a manually generated truthset of 1,350 tweets, performed two assessments tasks [10] for systematically assessing the quality of the tweet groups according to software practitioners' judgement and evaluated the ranking step against the assessment of software practitioners. Our results are encouraging and show that ALERTme is able to (1) detect tweets containing improvement requests with an encouraging accuracy, (2) generate reasonably coherent topics, and (3) create tweet rankings that strongly agree with practitioners' assessments.

IV. Conclusion

We described ALERTme, an approach to automatically *classify*, *group* and *rank* tweets for their use during software evolution. Our initial results show that applying mining techniques that are similar to those previously used on the lengthier user reviews from app stores is a promising direction. In future work we will conduct a more extensive evaluation and consider additional mining techniques for further purposes.

Acknowledgment

This work was partially supported by the European Comission within the SUPERSEDE project (ID 644018).

References

[1] E. Guzman, R. Alkadhi, and N. Seyff, "A Needle in a Haystack: What Do Twitter Users Say about Software?" in *Proc of the International Requirements Engineering Conference*, 2016, pp. 96–105.

[2] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, "AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace," in *Proc of the International Conference on Software Engineering*, 2014, pp. 767–778.

[3] L. V. Galvis Carreño and K. Winbladh, "Analysis of user comments: an approach for software requirements evolution," in *Proc of the International Conference on Software Engineering*, 2013, pp. 582–591.

[4] E. Guzman and W. Maalej, "How do users like this Feature? A fine grained sentiment analysis of app reviews," in *Proc of the International Conference on Requirements Engineering*, 2014, pp. 153–162.

[5] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc of the International Conference on Machine Learning*, 2006, pp. 161–168.

[6] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proc of the International Conference on World Wide Web*, 2013, pp. 1445–1456.

[7] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *Proc of the International Conference on Web Search and Data Mining*, 2011, pp. 65–74.

[8] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.

[9] E. Guzman, M. Ibrahim, and M. Glinz, "A Little Bird Told Me: Mining Tweets for Requirements and Software Evolution, submitted for publication."

[10] J. Chang, S. Gerrish, C. Wang, and D. M. Blei, "Reading Tea Leaves: How Humans Interpret Topic Models," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., 2009, pp. 288–296.