Poster: Automated User Reviews Analyser

Adelina Ciurumelea, Sebastiano Panichella, Harald C. Gall
University of Zurich, Department of Informatics, Switzerland
ciurumelea@ifi.uzh.ch,panichella@ifi.uzh.ch,gall@ifi.uzh.ch

ABSTRACT

We present a novel tool, AUREA, that automatically classifies mobile app reviews, filters and facilitates their analysis using fine grained mobile specific topics. We aim to help developers analyse the direct and valuable feedback that users provide through their reviews, in order to better plan maintenance and evolution activities for their apps. Reviews are often difficult to analyse because of their unstructured textual nature and their frequency, moreover only a third of them are informative. We believe that by using our tool, developers can reduce the amount of time required to analyse and understand the issues users encounter and plan appropriate change tasks.

KEYWORDS

Mobile Applications, User Reviews, Text Classification

1 INTRODUCTION

Mobile applications are highly popular, Google Play and the Apple Store host more than 2 billion apps each and enable the download of millions of apps every day. Mobile marketplaces allow users to provide direct feedback to developers through reviews. Nevertheless, these are difficult to analyse as (i) they consist of unstructured text with a low descriptive quality (ii) only a third of them are actually informative [1, 8] and (iii) popular apps can receive up to several thousands of reviews per day [9]. Researchers studied the characteristics of user comments and observed that they include bugs and feature requests [9], experience descriptions with specific features [6], feature enhancements requests [7] and comparisons with other apps. Several approaches [1, 4–7, 11] have been proposed for automatically classifying reviews according to a restricted set of classes: as *informative and non-informative* or as *feature request, bug and other* and then clustering them based on textual similarity, but this results in unstructured groups of reviews that have to be manually analysed and understood by developers in order to extract meaningful change tasks.

In this extended abstract, we present a tool prototype that aims to support the continuous integration of user feedback in the maintenance process through the automatic classification, filtering and analysis of reviews according to the fine grained mobile specific topics described in Table 1. The main contributions are: (i) a dataset of 6107 reviews manually labeled according to the categories defined in Table 1, (ii) the open source implementation of the tool [2] and a (iii) quantitative and qualitative evaluation.

2 APPROACH OVERVIEW

Our tool prototype, AUREA, uses pre-trained Machine Learning models to classify app reviews according to the fine grained topics specified in Table 1. The details about how the taxonomy was developed are included in our previous work [3].

We collected a set of 6107 app reviews from 37 open source apps available on Google Play, the apps belong to different categories and were carefully chosen to ensure a varied vocabulary. One author of the paper manually labeled this dataset which was then later used for training a distinct ML model for each category from our taxonomy. Before the training we applied for each review text a preprocessing and a feature extraction step. The preprocessing included punctuation and stop words removal and applying the Snowball Stemmer to reduce words to their root form. As features we extracted the tf-idf scores of the unigrams, bigrams and trigrams of the preprocessed texts. We used the Gradient Boosted Trees implementation from the scikit-learn [10] library as ML models.

Here we would like to note that users often address multiple topics in a single review, therefore it was necessary to perform multi-label classification, that is for a specific review return the list of matching categories. We achieved this by training a separate classifiers for each category. For example for the review "*On Marshmallow, the screen is buggy and sometimes shows the notification shade*", our tool can return the following categories: *Android version*, *UI* and *Complaint*. This is different from previous work that often performs single-label classification on review sentences. Classifying single sentences has several drawbacks: sentences taken out of context make the reported issue harder to understand. Additionally reviews are often grammatically incorrect, therefore sentence splitting is likely to be error prone and it is still possible that users will include several issues in a single sentence.

3 USAGE SCENARIO

AUREA provides an intuitive and user-friendly web interface that allows developers to easily upload, analyse and filter their reviews based on the pre-defined categories from Table 1.

The tool supports the following usage scenario: the developer first uploads a csv file containing the reviews downloaded from Google Play. Next they can visualise the occurrences of the different topics and understand which ones are most often associated with *Complaints* as in Figure 1. For each category, the percentage and absolute number of reviews that are not classified as *Complaints* (in

Table 1: Taxonomy of Review Categories

Review Category	Description
Device	mentions a specific mobile phone device (*i.e. Galaxy 6*).
Android Version	references the OS version. (*i.e. Marshmallow*).
Hardware	talks about a specific hardware component.
App Usability	talks about ease or difficulty in using a feature.
UI	mentions an UI element (*i.e. button, menu item*).
Performance	talks about the performance of the app (*slow, fast*).
Battery	references related to the battery (*i.e. drains battery*).
Memory	mentions issues related to the memory (*i.e. out of memory*).
Licensing	references the licensing model of the app (*i.e. free, pro version*)
Price	talks about money aspects (*i.e. donated 5$*).
Security	talks about security/lack of it.
Privacy	issues related to permissions and user data.
Complaint	the users reports or complains about an issue with the app
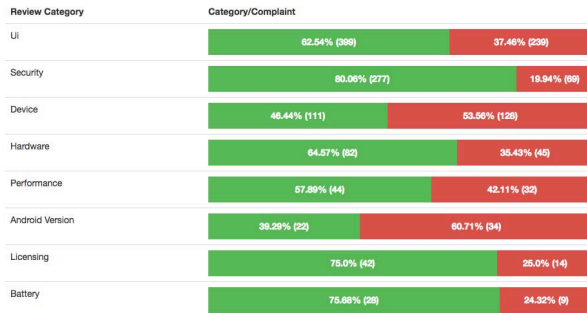
Figure 1: Analysis Results for the AcDisplay App.

green) and the ones that are (in red) are shown, thus the developer can quickly notice which categories are problematic for the users, *e.g.* in Figure 1 it is easy to see that the most troublesome category is *Android version*. Subsequently, the developer can decide to only look at reviews belonging to this category and understand what problems the users are reporting. For example, the developer might filter the reviews further and look at only the ones that are classified as *Android version* and *Performance*. Another use case is for the developer to focus on each mobile specific topic and understand if the developers report problems (*Complaint*) or are in general happy with that particular aspect of the application.

Figure 2: Classification Results for the AcDisplay App.

4 EVALUATION

We evaluated our tool prototype from both a quantitative and preliminary qualitative perspective. We analysed the performance of the ML models using a 10-fold cross-validation approach and we obtained an average precision of 0.836, an average recall of 0.763

and an average F1 score of 0.793. Our tool shows good accuracy for automatically classifying reviews according to the fine grained taxonomy we defined. Finally we asked three external evaluators to analyse the reviews of 3 open source apps using our tool and Excel and report their experience. Overall the feedback from the preliminary user study was positive, the evaluators reported that they needed less time to analyse the reviews using our tool as opposed to Excel and that they found it *helpful*. The full details of the evaluation are available in the online repository of the tool.

5 CONCLUSIONS

We present a novel tool, AUREA, that is able to classify and filter mobile apps reviews based on fine grained mobile specific categories. Through our work we want to help developers analyse the reviews of their apps in less time, better comprehend what issues users are reporting and plan their change tasks accordingly. The evaluation showed that our tool obtains high precision and recall in classifying reviews and our study participants found it helpful for analyzing reviews. We plan as future work to extend our dataset, experiment with different summarisation techniques and conduct a more comprehensive qualitative evaluation.

ACKNOWLEDGMENTS

REFERENCES

[1] N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, and B. Zhang. Ar-miner: Mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, pages 767–778, New York, NY, USA, 2014. ACM.

[2] A. Ciurumelea. URR. https://github.com/adelinac/urr/. [Online; accessed 31-January-2018].

[3] A. Ciurumelea, A. Schaufelbühl, S. Panichella, and H. C. Gall. Analyzing reviews and code of mobile apps for better release planning. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 91–102, Feb 2017.

[4] A. Di Sorbo, S. Panichella, C. V. Alexandru, J. Shimagaki, C. A. Visaggio, G. Canfora, and H. C. Gall. What would users change in my app? summarizing app reviews for recommending software changes. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2016, pages 499–510, New York, NY, USA, 2016. ACM.

[5] E. Guzman, M. El-Haliby, and B. Bruegge. Ensemble methods for app review classification: An approach for software evolution (n). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 771–776, Nov 2015.

[6] E. Guzman and W. Maalej. How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 153–162, Aug 2014.

[7] C. Iacob and R. Harrison. Retrieving and analyzing mobile apps feature requests from online reviews. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 41–44, Piscataway, NJ, USA, 2013. IEEE Press.

[8] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman. A survey of app store analysis for software engineering. *IEEE Transactions on Software Engineering*, PP(99):1–1, 2017.

[9] D. Pagano and W. Maalej. User feedback in the appstore: An empirical study. In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pages 125–134, July 2013.

[10] scikit-learn developers. Gradient Boosting Classifier. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html. [Online; accessed 14-June-2017].

[11] L. Villarroel, G. Bavota, B. Russo, R. Oliveto, and M. Di Penta. Release planning of mobile apps based on user reviews. In *Proceedings of the 38th International Conference on Software Engineering*, ICSE '16, pages 14–24, New York, NY, USA, 2016. ACM.