2014 Second IEEE Working Conference on Software Visualization

FAVe: Visualizing User Feedback for Software Evolution

Emitza Guzman
Technische Universität München
Garching, Germany
emitza.guzman@mytum.de

Padma Bhuvanagiri
Technische Universität München
Garching, Germany
padmabhuvanagiri@mytum.de

Bernd Bruegge
Technische Universität München
Garching, Germany
bruegge@in.tum.de

***Abstract*—App users can submit feedback about downloaded apps by writing review comments and giving star ratings directly in the distribution platforms. Previous research has shown that this type of feedback contains important information for software evolution. However, in the case of the most popular apps, the amount of received feedback and its unstructured nature can produce difficulties in its analysis. We present an interactive user feedback visualization which displays app reviews from four different points of view: general, review based, feature based and topic-feature based. We conducted a study which visualized 2009 reviews from the Dropbox app available in the App Store. Participants considered the approach useful for software evolution tasks as they found it could aid developers and analysts get an overview of the most and least popular app features, and to prioritize their work. While using different strategies to find relevant information during the study, most participants came to the same conclusions regarding the user reviews and assigned tasks.**

I. Introduction

Application distribution platforms, or app stores, allow users to share their opinion about downloaded apps through text reviews, where they can, e.g., express their satisfaction with a specific app feature, request a new feature or report a bug. The reviews can be used to drive the development effort and improve future releases.

However, review analysis requires an extensive human effort due to the *large amount* of reviews and the *unstructured nature* of its textual content. These challenges can prevent analysts and development teams from using the information in the reviews during the app evolution.

To reduce the user feedback analysis effort we propose FeedbAck Visualization, FAVe, an approach to visualize user reviews on four different abstraction levels: general, review based, feature based and feature-topic based. FAVe contains rating and sentiment information and can help developers and analysts get an overview of the most and least popular app features, as well as the rating and sentiment distributions among the reviews. Ratings generally evaluate apps on a general level. Therefore, we use lexical sentiment analysis [9] to analyze the text in a finer-grained level and detect the opinions users have about specific mentioned app features. A collocation finding algorithm [7] is used for extracting the mentioned features. Additionally, we apply topic modeling [2] for grouping related features and further reducing the information overload. FAVe's interactive nature allows for the navigation of different review granularities: from groups of features, to single features, to the actual review text that contains the features. Furthermore, different filters allow FAVe users to customize the amount of displayed information. While previous research work does not usually focus on the display of actual review text [1], we consider it a crucial piece of information in software evolution as it can help developers and analysts find the reasons behind app feature popularity or lack thereof and aid them in taking the appropriate measures to address current issues when necessary.

II. Mining User Feedback

To generate the data displayed by FAVe we use Natural Language Processing and Data Mining techniques. The mining approach was described in previous work [5] and consists of four main steps. First, we *preprocess* the comment and title of each review and prepare it for feature extraction. Then, we apply a *collocation algorithm* and extract the mentioned app features. Afterwards, we apply *sentiment analysis* to the comment and title in the review and assign a sentiment to each extracted feature. Finally, we use *topic modeling* to group related features. In the following we explain the main steps of our mining approach.

A. Preprocessing

In this first step we extract the title and comment for each review. The sentiment analysis process does not require any additional steps. However, the feature extraction process requires that the following steps be executed (1) extraction of verbs, nouns and adjectives, (2) stopword removal, and (3) lemmatization.

B. Feature Extraction

We use the collocation finding algorithm provided by the NLTK[1] toolkit for extracting features from the user reviews. A collocation is a collection of words that co-occur unusually often within a certain word distance. Examples of collocations which describe features in user reviews are the set of words <*pdf viewer*> and <*user interface*>. We use the likelihood-ratio test as the criteria for finding collocations of two word length in our reviews. We regard word ordering as unimportant for describing features e.g. the collocations <*picture view*> and <*view picture*> are grouped into the same collocation.

[1]http://www.nltk.org/

978-0-7695-5305-4/14 $31.00 © 2014 IEEE
DOI 10.1109/VISSOFT.2014.33

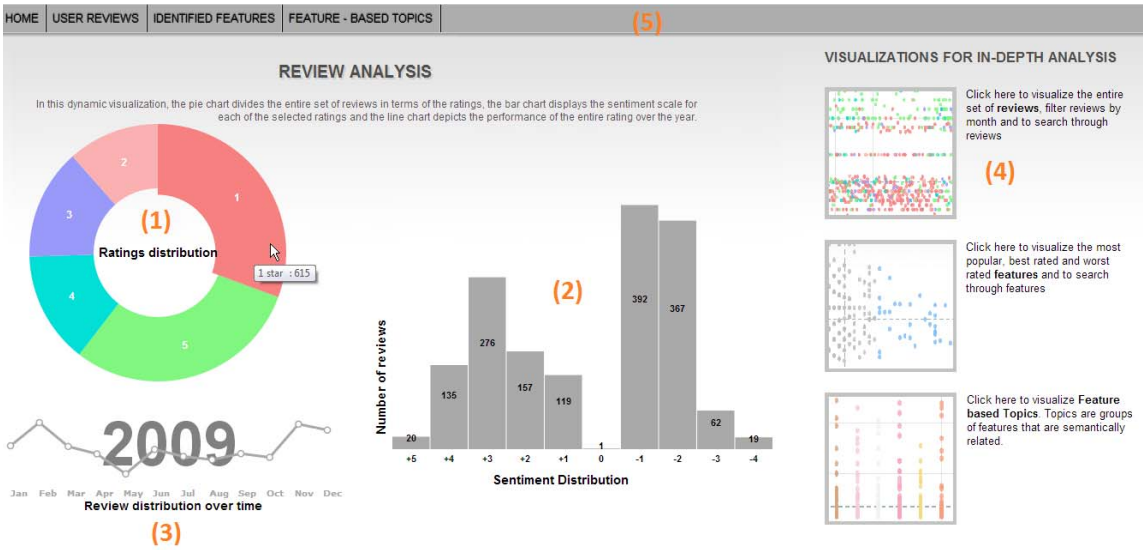

Fig. 1: Home screen view. The following aspects are shown in the view: (1) Rating distribution, (2) Sentiment distribution (2) Review distribution over time, (4) Visualizations for finer-grained analysis, (5) Navigation menu.

C. Sentiment Analysis

We use the lexical sentiment analysis tool SentiStrength [9] for finding users' opinions about features. SentiStrength divides the review text into sentences and then assigns a positive and negative value to each sentence. The positive scores are in the (1, 5] range, whereas the negative scores are in the (-1,-5] range. The [1,-1] range is used for neutral sentiments. The sentiment score of the whole sentence is computed by taking the maximum and minimum scores among all the words in a sentence. We compute the sentiment of an entire review by calculating the positive and negative average scores of all sentences in the review separately. For the case where both positive and negative sentence averages are in the [-1,1] range we assign the whole review the neutral score of 0. When the absolute value of the review negative average multiplied by 1.5 is larger than the positive average of the review, we assign the review the sentiment score of the negative average[2]. In the opposite case, the review is assigned the positive average score. We assign a feature the sentiment score of the review where it is located.

D. Topic Modeling

We group features that tend to co-occur in the same reviews by using Latent Dirichlet Allocation (LDA) [2], a topic modeling algorithm. In LDA a topic is a probabilistic distribution over words and each document is modeled as a mixture of topics. This means that each review can be associated to different topics and that topics are associated to different words with a certain probability. We used the Matlab Topic Modeling Toolbox[3] implementation for our approach. Instead of using the words forming the vocabulary of our analyzed reviews in the LDA algorithm, we input the list of extracted features and model each feature as a single word. For example, the feature described with the *<picture view>* collocation is transformed into the single term $picture\_view$. LDA then outputs the feature distribution of each topic and the probabilistic topic composition for each review. An example of a topic with this modification could then be the set of features *{picture_view, camera_picture, upload_picture, delete_picture}* which describes features related to manipulating pictures in an application.

[2]As explained in the SentiStrength user manual: http://sentistrength.wlv.ac.uk/

[3]http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

III. Visualizing User Feedback

FAVe has two main components: (1) a *home screen* which shows an overview of the reviews, its ratings and the sentiments expressed in the reviews and (2) *fine-grained analysis visualizations* which allow for a more detailed analysis by interactively navigating different abstractions levels of reviews, mentioned features and groups of features. We used the D3.js library[4] for the implementation of the visualization prototype. In the following sections we describe the two main components of FAVe, the possible interactions and the coloring scheme used.

A. Home Screen

The home screen of FAVe is a simple interactive dashboard. It provides a dynamic visualization of the user reviews in terms of star ratings, user sentiment associated with each review and a cumulative rating performance over the entire year. Figure 1 shows the home screen of FAVe, which contains four essential components:

[4]http://d3js.org/

1) Rating distribution: The interactive pie chart shows the overall distribution of the app's ratings, in terms of the number of stars given in the user reviews. When clicking on the different ratings shown in the pie chart, the rest of the graphs in the home screen are updated to reflect the information about the selected pie chart rating.

2) Sentiment distribution: When no type of rating is selected in the rating distribution pie chart, the sentiment bar graph is displayed in a dark grey color, depicting the overall user sentiments of all reviews. When the reviews with a particular type of rating are selected from the ratings distribution pie chart, the sentiment bar graph automatically changes to display the sentiment scale of the selected reviews, changing its color to the one belonging to the selected rating.

3) Review distribution over time: The line graph shows the month-wise distribution of all reviews. When the visualization user chooses a particular rating in the rating pie chart, this graph dynamically changes to display the month-wise distribution of the reviews of the selected rating.

4) Fine-grained visualizations overview: This component provides an overview of the three different types of finer-grained user feedback views: review based, feature based and feature-topic based. Hovering the mouse over each image, enlarges it, allowing the user to get a more detailed view. We explain more about each fine-grained visualization and its possible interactions in the next section.

B. Fine-grained Visualizations

FAVe has three visualizations for fine grained analysis which will be explained in the following sections.

1) Review based: This interactive visualization provides detailed information about the app reviews' distribution over time. It captures two main aspects: the sentiment score of each review and its rating. The reviews are visualized as hexagonal points in a scatter plot. The y-axis of the scatter plot depicts the sentiment score of the reviews and the review-points are color-coded to reflect the ratings. For popular apps the number of reviews received from the customers is generally in the order of thousands or more. Visualizing all of them in a single scatter plot can be overwhelming for the user, as the graph seems over-crowded. In order to reduce information overload only reviews with automatically extracted features are displayed, as we consider this reviews to be the most informative for developers, e.g. general praise or complaints are of less interest for software evolution tasks. Further filters, explained in section III-C, allow FAVe's users to further reduce information overload in this view. Figure 2 shows the review based visualization displaying the reviews with the highest ratings for the January and February months.

2) Feature based: This visualization shows the average sentiment score and appearance frequency of each of the extracted app features. The features are visualized as hexagonal points in a scatter plot. In the plot, each hexagonal point representing an identified feature serves as a link to visualize all the underlying reviews that have comments concerning the feature. Therefore, when users clicks on a point, a scatter plot depicting all the reviews mentioning the clicked feature is shown.

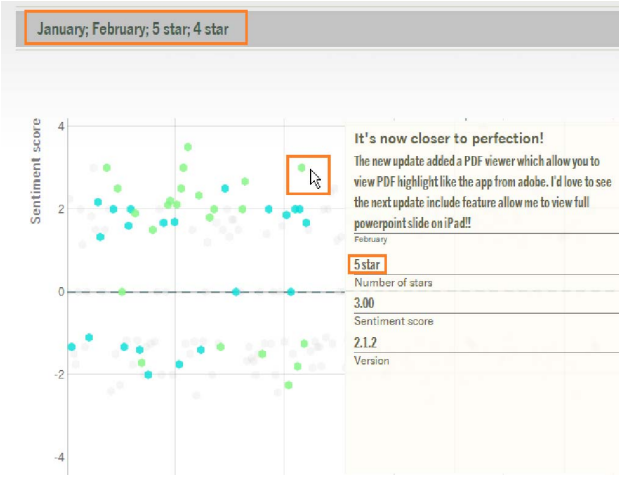

Fig. 2: Hovering over a point the the review based view where only some of the months are activated.

3) Feature-topic based: This visualization shows groups of related features. The features in each group topic are visualized as hexagonal points in a scatter plot. The y-axis of the plot depicts the frequency of each feature and the x-axis depicts the different topics. Each topic is depicted in a unique color and named after the most frequent feature in the topic. As in the feature based visualization, each hexagonal point representing an identified feature in the scatter plot serves as a link to visualize all the underlying reviews that mention the feature.

C. Interactions with Fine-grained Visualizations

There are four main interactions in FAVe:

1) Zoom and Pan: The three scatter plots of the fine-grained visualizations are enabled with both zoom and pan features. Double-clicking at any point in the graph allows the user to zoom into the scatter plot. Additionally, the user can drag or pan the mouse to shift the visualization component to another screen area.

2) Detailed Information Display: Hovering over each point in the fine-grained visualization's scatter plot enables the user to view: 1) *review fine-grained details* such as the review title, comment, number of stars, sentiment score, app version number and the date in which the review was written or 2) *feature fine-grained details* such as its frequency, positive score and negative score. We visualize the positive and negative scores in order to avoid loosing important information due to averaging [4] and to aid developers and analysts detect conflicting opinions about certain features. Additionally, when pointing to the different parts of the home screen ring chart and line graph further information about each rating or time point is displayed through a tooltip, an example of this tooltip can be seen in Figure 1 next to the mouse pointer.

3) Keyword Search: All fine-grained visualizations include a search box. In the search box users can enter multiple words.

The displayed results are then filtered to only contain the entities containing the words typed into the search box or its lemmas.

4) Information Filtering: To reduce information overload users can choose to only visualize features which frequency is higher than a given threshold, or to visualize reviews that only mention features that are mentioned at least *N* given times. Furthermore, in the review based view a dropdown menu offers a filter for pruning reviews month-wise, as well as based on rating.

D. Coloring Scheme

Except for the topic based visualization, where each topic is depicted in a unique color. All visualizations in FAVe are color coded to reflect the ratings. Red is used for the lowest rating (1 star), while green is used for the highest rating (5 stars). The intermediary colors pink, purple and blue, reflect the intermediary ratings (2-4 stars).

IV. Preliminary Study

We evaluated the usability of FAVe by conducting a user study with 5 software developers. All participants were in the information technology industry and had an industry experience between 1 and 4 years, with an average experience mean of 2.8 years. Their roles were varied, two of them were system engineers, whereas one was a quality engineer, web developer and database administrator. Four participants reported having previous experiences as technical consultants, human-machine interface designers and web developers. Two of the user study participants were female and three were male.

For the study, all of the Dropbox app reviews available in the App Store for the year 2013 were visualized. A total of 2009 reviews and 600 unique extracted features were input to FAVe.

At the beginning of the study one of the authors introduced the Dropbox app and the participants were shortly briefed about FAVe, its main views and the possible interactions. Afterwards, the participants had 6 to 7 minutes for interacting and exploring the tool as they wished. Next, each participant was given two tasks in which they had to imagine they were developers working for the Dropbox app. In the first task, participants had to detect the three most urgent issues based on the user review comments and asked to justify their choices. In the second task, they were asked about the general user opinion of the *pdf viewer* feature. Additionally, they were asked to identify if there were conflicting opinions concerning the *pdf viewer* feature and to identify which other features users frequently mentioned when writing about the aforementioned feature. During the execution of the tasks one of the authors observed each participant and took note of the interactions done with the tool and the participant's comments.

A. Identifying Urgent Issues

Participants used two different strategies for identifying the three most urgent issues. Three participants used the review based view, whereas two participants used the feature based view. Only one participant analyzed the home screen pie chart to get an idea of the number of negative reviews before navigating to the review based view. Participants using the review based view followed a similar workflow: they filtered the reviews from the most recent months, and lower ratings. Additionally, they applied a frequency filter in the visualization so that only reviews with popular features would be shown. Afterwards, they only focused on the reviews with lower sentiment. All of the participants using the review based visualization navigated to the actual review text. The participants using the feature based approach concentrated on the most frequently mentioned features with the most negative sentiments. One of the participants navigated to the actual review text, whereas the other identified the features without looking for further information. Since the current version of FAVe does not contain any additional filters in the feature based view, none of the participants reduced the shown information.

Independently of the used strategy, we found that participants agreed in most of the issues identified as urgent. Two participants paid special attention to the version information in the reviews' detailed view, indicating that this is important information for some developers. Interestingly, while participants asked how the sentiments in the reviews were computed, none of the participants looked at the sentiment scores displayed in the actual review text while performing their tasks, but rather at the sentiment quadrants were the points were displayed. This could be an indicator that actual sentiment scores are very fine-grained information.

B. Identifying General Opinions, Conflicting Opinions and Co-occurring Features

In the second task, participants used varied strategies. Two participants used a combination of the feature and topic based views for solving the task. Whereas, one participant used the single review based, feature based and topic based views. Two possible explanations for the variety of used strategies can be the different information processing tactics of each participant or their unfamiliarity with FAVe. However, independently of the used strategy, all participants found that the *pdf viewer* feature had conflicting user opinions and found similar sets of co-occurring features, with the exception of one participant who when analyzing the single reviews' text declared that no additional features were being mentioned when writing about the *pdf viewer* feature.

C. General Feedback

After the execution of the two tasks, participants were asked about the perceived usefulness of FAVe for those involved in software development, about the amount of information and levels of granularity displayed by the tool, as well as for comments or improvement suggestions. All participants thought that the tool would be helpful for developers and others involved in software development, such as testers and people from quality assurance. One participant thought FAVe would also be useful for end app users. Additionally, all participants answered affirmatively when asked if they would use the tool for their

work if available. They thought that the tool could allow them to identify issues and prioritize their tasks. Furthermore, one of the participants praised FAVe for displaying the actual review text. Two participants mentioned that a particular weakness of the tool was based on its display of user reviews, without any previous user filtering. On this respect one participant commented: *The usefulness of this tool depends on the quality of reviews because at times the users can be exaggerated and biased*, whereas another participant mentioned: *The people from whom the reviews are considered matters a lot. They have to be focused on a subset of people who can give honest and useful reviews.* All participants said that the amount of information displayed in the study was manageable and that the filters were very useful for reducing the information and finding what they were interested in. Furthermore, all participants agreed that the tool had a learning curve and that some of the main components (topics and sentiments) needed an explanation because they were not familiar with the terms. During the study participants were interested in understanding the cases were there was a mismatch between the rating and the sentiment score. Some of these cases were because of limitations in the sentiment analysis, while others were due to the neutral language used in the review. Only one of the participants mentioned that she would wish for a higher quality in the naming of the features, indicating that the users were satisfied with the feature extraction mechanism.

V. Related Work

To the best of our knowledge no previous research has explored the visualization of user reviews for software evolution. However, user feedback visualization has been an active research topic in other domains.

Liu et al. [6] visualized positive and negative opinions of product (i.e. printers and cameras) features with bar charts.The main differences between their approach and FAVe is the interactiveness of the visualization and the different levels of granularity that FAVe offers. OpinionBlocks [1] is an interactive visualization which displays increasingly detailed textual information from user reviews. The information provided by the visualization is based on manually extracted and grouped features, as well as their sentiments, while FAVe bases the visualization on automatically extracted information, displaying additional attributes such as time and rating, as well as enabling the search and visualization of targeted information. OpinionSeer [10] visualizes features and sentiments extracted from hotel reviews. The authors use a radial visualization to compare the mentioned features and their associated sentiments against different user demographics. Our visualization approaches are complementary and FAVe could benefit from demographic visualizations to aid developers and analysts in understanding app users and their diverse needs. Oelke et al. [8] visualized features and their associated sentiments for printer reviews. The main difference with FAVe is FAVe's focus on a single app, as well FAVe's display of actual review text in the most detailed views. Opinion Space [3] is a visualization tool which offers several alternatives which allow end users to navigate review comments which are diverse in terms of the opinions and experiences expressed in the review. FAVe could be complemented by visualizations where conflicting opinions concerning app features are displayed, helping developers and analysts detect and reason about conflicting opinions, as well as make appropriate decisions, i.e. creation of different software product lines.

VI. Conclusions and Future Work

We presented FAVe, a visualization tool for analyzing app user feedback in terms of the received ratings, mentioned app features and expressed sentiments. The visualization allows for a finer-grained analysis than the one offered by traditional review ratings and allows for an interactive in depth analysis of user reviews, where users can navigate from a general overview, to groups of similar features, single features and the actual text of the reviews. FAVe avoids information overload by only displaying reviews which contain automatically extracted features and offering time, rating and feature popularity filters. In the future we plan to enhance the visualization to allow analysts and developers detect conflicting opinions, as well as to analyze user reviews with respect to the users' demographic characteristics. The preliminary study shows that participants have different strategies for acquiring user feedback information, suggesting the importance of the different granularity views in FAVe. Furthermore, the abstracted information tended to be similar for all FAVe users, independently of their used tactic. A more extensive study will help to further analyze the strengths of each strategy and determine the helpfulness of FAVe for software evolution.

References

[1] B. Alper, H. Yang, E. Haber, and E. Kandogan. OpinionBlocks: Visualizing Consumer Reviews. In *Proceedings of the IEEE VisWeek Workshop on Interactive Text Analytics for Decision Making*, 2011.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[3] S. Faridani, E. Bitton, K. Ryokai, and K. Goldberg. Opinion space: a scalable tool for browsing online comments. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, page 1175. ACM Press, Apr. 2010.

[4] E. Guzman, D. Azócar, and Y. Li. Sentiment analysis of commit comments in GitHub: an empirical study. In *Proceedings of the 11th Working Conference on Mining Software Repositories - MSR 2014*, pages 352–355, New York, New York, USA, May 2014. ACM Press.

[5] E. Guzman and W. Maalej. Do Users Like this Feature? A Fine Grained Sentiment Analysis of App Reviews. In *Proc. of the International Conference on Requirements Engineering - RE '14, to appear*, 2014.

[6] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the Web. In *Proceedings of the 14th international conference on World Wide Web - WWW '05*, pages 342–351. ACM Press, May 2005.

[7] H. Manning, Christopher D., Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[8] D. Oelke, M. Hao, C. Rohrdantz, D. A. Keim, U. Dayal, L.-E. Haug, and H. Janetzko. Visual opinion analysis of customer feedback data. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 187–194. IEEE, 2009.

[9] 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*, 61(12):2544–2558, Dec. 2010.

[10] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu. OpinionSeer: interactive visualization of hotel customer feedback. *IEEE transactions on visualization and computer graphics*, 16(6):1109–18, Jan. 2010.