University of Tartu

Institute of Computer Science

Department of Software Engineering

Bejon Sarker

A literature review: Automatic Extraction of app features from user reviews

Seminar on Enterprise Software, 2018
Abstract

Smart devices have grown its popularity in past decade. Now App distribution platform such as Apple Play store, Google play store contains millions of applications. Users can review all those applications and can also rate them. This enormous amount of data contains useful information for both developer and Play store participant. Reviews contain user sentiment and opinion. Mining all this information can extract various useful features of the application such as new feature request, bug related to the specific feature. So, mining reviews could be beneficiary for requirement engineering.

This growing number of apps and reviews has accelerated researchers interest in mining data from reviews. Some academic studies are conducted for extracting app features from app reviews. The objectives of the study are to identify the techniques and methods used to extraction important information which can improve app developer insight about users’ acceptance. Moreover, the study focuses on the comparative accuracy and domain of this methods. The future research goal and industry direction are also discussed at the end of the research depending on the data evaluated.
Table of Contents

Introduction ................................................................................................................... 4
Search Approach ........................................................................................................ 5
  Planning the Study ..................................................................................................... 5
  Performing the Review .............................................................................................. 7
  Reporting the Review ............................................................................................... 8
Result ............................................................................................................................ 9
Discussion ................................................................................................................... 12
  Challenges ................................................................................................................ 12
  Future work and industry direction ......................................................................... 12
References .................................................................................................................. 13
1. Introduction

Smartphones and tablets are a ubiquitous part of our daily life. Popularity of these Apps is increasing in recent years. These applications are generally downloaded from application distribution platforms like the Google Play Store, Apple App Store or the Windows Store. These application distribution platforms also allow the users to provide review and rate for assessing applications. Users provide their opinion about different aspects of the applications such as bug reporting, new feature request, positive reviews, etc. These reviews are important to the developers to know what the users’ sentiment about the app is, what part of the app users like or dislike the most. But some applications have an enormous amount of reviews and examine them manually is a very tedious work. As an example, 22 reviews are submitted per app per day by IOS users and Facebook receives more than 4000 reviews per day (Guzman & Maalej, 2014).

Over recent years, various automated techniques have been proposed by researchers to extract and analyze user sentiments and opinions from app reviews. The first RQ (Research Question) of this study is to reveal the techniques and procedures that are used by previous researchers to extract different app features from the reviews. Previous research was performed on a different set of data. Fu et al. (2013) mined total 13,286,706 reviews of 171,493 apps of Google play store whereas Guzman and Maalej (2014) mined total 32210 reviews of 7 apps of both Google play store and Apple play store. So, the accuracy of these research was not being evaluated comparatively on a common set of data. The second research question desires to evaluate the accuracy of these different techniques.

The goal of this study is to find out the methods used to extract app features and user sentiment about those features which will help developers to find the answer of the question: which part of the app is liked or disliked by the users? The study will also provide the accuracy comparison between different methods and their context. Method context is necessary because reviews meaning in one type of app may have very different meaning in other types of app.
The structure of this study as follows. Next part describes search approach with research questions. Section 3 describes the result of the study. And last section presents the discussion on challenges and future work.

2. Search Approach

The study follows the guidelines specified by (Kitchenham, 2004). The Literature Review is structured into three phases.

1) Planning of the study
2) Performing the review
3) Reporting the review

The overall process is described in Fig 1.

2.1. Planning of the study

This phrase starts with specifying the research questions and the inspirations followed by developing review protocol for search questions. At the end of planning part, the study validates the designed review protocol.

2.1.1. Research Questions

**RQ1:** What procedures or techniques are used to extract features and sentiments from reviews on play stores?

**Inspiration:** Application Distribution Platforms such as Google Play Store or Apple play store contains numerous information of user opinion and sentiment in the form of reviews. These reviews vary depending on the category of the application. Reviews also have some characteristics such as review sentences are not complete, sentences may not be grammatically correct. Moreover, reviews are not formal and shorter than usual text which makes it even harder to extract meaningful information from it. For the past few years Researchers have proposed various data mining approaches to extract features and user sentiment from reviews. Most of the techniques are based on supervised learning. This question’s target is to find out these different data mining techniques that are proposed by previous researchers to extract feature information from the app reviews.

**RQ2:** How accurate the methods are depending on the targeted domain of that method?

**Inspiration:** App reviews (voice of customers) are very useful for developers or tester to figure out likes and dislikes of users. These reviews may contain new feature request, one specific bug report which can help developers or testers to improve the application in future. There are various techniques to extract features from user reviews but all these methods are widely dependent on the domain of the application. One method
Fig 1. SLR Process

Phase 1: Plan review

1. Specify research questions
   2. Develop review protocol
   3. Validate review protocol

Phase 2: Conduct review

4. Identify relevant research
   5. Select primary studies
   6. Assess study quality
   7. Extract required data
   8. Analyze data

Phase 3: Document review

9. Write review report
may have good accuracy in one domain which will perform badly if it is applied in another domain. This research question aims to evaluate the accuracy of these data mining techniques in different domain. It will help developers to select the most accurate technique for mining reviews for their application.

### 2.1.2 Review Protocol

I have used following libraries for searching literatures.

1. IEEExplore
2. Springer Link

Given search string was prepared to conduct searching in these libraries. I also checked reference list to find out secondary studies if any related study was missed out.

**Search String**: ((user OR (review OR comment OR text)) OR (sentiment OR opinion) OR rating OR (analysis OR mining OR data-mining OR processing) OR (requirement OR feature) OR request OR (bug OR issue OR complain OR report) OR (accuracy OR reliability)) AND (mobile OR ((application OR applications OR app OR apps) OR (Playstore OR store OR market OR AppStore)))

### 2.2. Performing the Review

**Selection procedure**: Study was selected based on following procedure:

1) First stage filtering was based of the title and abstract of the papers. If one paper does correlate with the topic and research question I excluded that.

2) At this stage I read the introductions and those papers didn’t correlate I excluded them.

3) I filtered the remaining based of table 1.

4) I checked the reference list to find out the secondary studies which are not found by the search string.

I checked the quality of the papers by following Keele (2007). See table 2.
Table 1
Inclusion and Exclusion Criteria

<table>
<thead>
<tr>
<th>Inclusion Criteria</th>
<th>Exclusion Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Analysis of feature extraction from reviews should be provided.</td>
<td>1. Must meet the quality checklist of table 2.</td>
</tr>
<tr>
<td>2. Proper accuracy evaluation of the procedure mentioned.</td>
<td>2. No proper accuracy evaluation is specified.</td>
</tr>
<tr>
<td>3. Clear definition about the data category is specified.</td>
<td>3. Are not correlated with the research questions.</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the study contain clearly defined aim?</td>
</tr>
<tr>
<td>Does the study explain the data collection procedure properly?</td>
</tr>
<tr>
<td>Is the study replicable?</td>
</tr>
<tr>
<td>Does the study explain threads to the validity adequately?</td>
</tr>
<tr>
<td>Does the study contain clean reporting?</td>
</tr>
<tr>
<td>Does the study provide background of accuracy evaluation?</td>
</tr>
</tbody>
</table>

After following all the procedures and examining the literature I finally got 6 studies.

2.3. Reporting of the Review

Data was extracted from the 6 studies to answer the previously specified research questions. The next section provides the reporting of the collected information and answers to the research questions.
3. Result

3.1. RQ1: What procedures or techniques are used to extract features and sentiments from reviews on play stores?

<table>
<thead>
<tr>
<th>Number</th>
<th>Title</th>
<th>Reference</th>
<th>Features extracted</th>
<th>Methods/ procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What parts of your apps are loved by users?</td>
<td>(Gu &amp; Kim, 2015)</td>
<td>Bug Report, New Feature Request, Praise, Feature Evaluation</td>
<td>POS tag, SDG, Parsing Tree</td>
</tr>
<tr>
<td></td>
<td>Maintenance and Evolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>How Do Users Like This Feature? A Fine Grained, Sentiment Analysis of</td>
<td>(Guzman &amp; Maalej, 2014)</td>
<td>Features related to the functionalities of the app</td>
<td>Latent Dirichlet Allocation (LDA)</td>
</tr>
<tr>
<td></td>
<td>App Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>AR-Miner: Mining Informative Reviews for Developers from Mobile App</td>
<td>(Chen, Lin, Hoi, Xiao, &amp; Zhang, 2014)</td>
<td>Topics of app can be identified. As example for Youtube app ‘button’, ‘theme’,</td>
<td>Expectation Maximization for Naive Bayes (EMNB), LDA,</td>
</tr>
<tr>
<td></td>
<td>Marketplace</td>
<td></td>
<td>‘keyboard’.</td>
<td>Aspect and Sentiment Unification (ASUM)</td>
</tr>
<tr>
<td>5</td>
<td>Facilitating developer-user interactions with mobile app review</td>
<td>(Oh, Kim, Lee, Lee, &amp; Song, 2013)</td>
<td>New non-functional request or Bugs related to the functionality of the app.</td>
<td>Support Vector Machine (SVM)</td>
</tr>
<tr>
<td></td>
<td>review digests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Why People Hate Your App – Making Sense of User Feedback in a Mobile</td>
<td>(Fu et al., 2013)</td>
<td>Overall sentiment like Stability, attractiveness, accuracy etc.</td>
<td>Statistical Analysis</td>
</tr>
<tr>
<td></td>
<td>App Store</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

App stores allows users to provide feedback about the app which is particularly very interesting for requirement engineering. These feedbacks analysis can help developers to improve their software product such as finding new feature and finding bugs related to some specific feature.
(Gu & Kim, 2015) proposes SUR-miner (Software User Review miner) which classifies reviews into five categories using pattern-based parser and evaluate the aspects of reviews. This tool can successfully parse praises, new feature requests, Bug reporting etc. The method is evaluated on 17 popular apps.

Panichella et al. (2015) presents a taxonomy to identify app reviews related to software evaluation and maintenance. The approach merges three methods: a) NLP (Natural Language Processing), b) Text Analysis and c) Sentiment Analysis.

Guzman and Maalej (2014) identifies the fine-grained features from review using NPL. The study also extracts user sentiments for each feature and prove them a general score. At last, by using topic modeling techniques fine-grained features are grouped into meaningful features. They evaluate their technique on 7 apps from Apple App store.

Chen et al. (2014) presents AR-miner (App Review miner). They structure there work in four steps. First the study extract reviews and filters the noise and irrelevance which is followed by grouping the reviews using topic modelling. Next, they prioritize the review and finally present the via intuitive visualization. They also performed a comparison between LDA (Latent Dirichlet Allocation) and ASUM (Aspect and Sentiment Unification) techniques on 4 different application and found that LDA performs worse.

Oh et al. (2013) presents a review analysis system using SVM technique which they tested on 24000 Google Play Store apps. This system can automatically categorize reviews into bug report, non-functional and functional feature requests categories.

Fu et al. (2013) propose a system (WisCom) which is able to identify inconsistent reviews and reasons why users like a app or dislike a app. The study provides insights into app market finding users major concerns and preferences. The study identifies in total 10 top factors. The study was conducted on in total 171,493 apps and 13,286,706 reviews.

### 3.2. RQ2: How accurate the methods are depending on the targeted domain of that method?

<table>
<thead>
<tr>
<th>Number</th>
<th>Title</th>
<th>Reference</th>
<th>Accuracy</th>
<th>Apps and review number</th>
<th>App Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What parts of your apps are loved by users?</td>
<td>(Gu &amp; Kim, 2015)</td>
<td>Precision: N/A, Recall: N/A, F1-score: 0.81</td>
<td>Apps: 17, Reviews: N/A</td>
<td>Productivity, Photography, Game, Social, Communication,</td>
</tr>
</tbody>
</table>
From the above table, an overall comparison of the performance between different methods are given. According to the F1-score Oh et al. (2013) has the best accuracy but the method doesn't extract any specific features of the app rather it focuses on only few aspects of the app like bugs related to the functionality of the app. Moreover, all these methods are not evaluated on a common set data. Such as Chen et al. (2014) only evaluated in 4 apps and how this method will perform on others app is totally unknown.
All the methods described here target only few domains of app and train their classifier algorithm only on that specific set of domains. So, the accuracy of all these methods are no evaluated on a common benchmark of dataset.

4. Discussion

App Stores contain tones of wealthy information about the users. App store participants and developer could user the useful information to understand the audience. Mining the reviews of application can provide developer an insight about users’ preferences like what new features they want or is there any bug related to specific feature. Similarly Mining the reviews of application can also help users to have better understanding of the application and what other users think about different features of the application. To date, there are only a limited number of researches are conducted in this field but the outcome is very promising. I am expecting more research on this same topic will be conducted in future. In this section, I will present the findings of the study.

4.1. Challenges

Challenges in extracting features from user reviews are listed below:

1. In most of the methods supervised or semi-supervised learning is used. Here, the threat is imbalance data. If on set becomes very imbalance then the accuracy of that method will hamper. Such as if one app receives very less reviews which are related to new feature request. Then it will be harder for the method to find new feature request correctly from the review. For some specific app this type of case is certainly possible.

2. Spam reviews hampers the accuracy of the method. In the whole set of reviews there will always be some spam reviews. But when the method is building its training dataset it is necessary to identify and filter all those spam reviews. Identification and filtering spam reviews is a challenge for extracting true features from the user reviews.

3. Cross-validation is not conducted and its difficult to perform. Relevance of the extracted features are not cross-validated.

4. How one specific method will perform out of its domain is unknown. The accuracy of one specific method cannot be evaluated out of its domain because review in different domain may be different meaning. Such as “unpredictable” in a movie or book review app may have good meaning where the same review may mean negative in a gaming app.

4.2. Future Research and Industry Direction

I envision that future research in this topic will be focused on finding one method that will reduce or remove the domain dependency. Which will be supervised one a large set of data and will be able to identify features of an app independent of the domain of the app.

I also envision that for analyzing user sentiment and opinion there will be various SaaS in the industry which will make it cheap a reasonable for developer to use. Developer will integrate their application with these kind of system and it will continuously
read the review and ratings and will keep the developer updated about the sentiment of their application.

References


