Can App Reviews Help Developers to Improve Mobile User Interface Design?

Wenge Le[1], Yong Wang[1,2], Cuiyun Gao[3], Liangfen Wei[4], Fei Yang[5]*

[1] School of Computer and Information, Anhui Polytechnic University, China
[2] Institute of artificial intelligence, Hefei Comprehensive National Science Center (Anhui Artificial Intelligence Laboratory), China
[3] School of Computer Science and Technology, Harbin Institute of Technology, China
[4] School of Information Engineering, ChaoHu University, China
[5] Zhejiang Lab, China
253902668@qq.com, yongwang@ahpu.edu.cn, cygao@cse.cuhk.edu.hk, 649489564@qq.com, yangf@zhejianglab.com

Abstract

For mobile user interface (M-UI) design, it has an important impact on app user's usage. However, M-UI design is limited by subjective factors, even professional developers can't determine whether the M-UI design is good or bad. App reviews provide an opportunity to proactively collect user complaints and promptly improve the user experience of apps. Therefore, it is meaningful to explore whether app reviews can help developers to improve M-UI design. In this article, we randomly select six different categories of apps from Google Play Store and App Store, with over 160000 reviews, and conduct a preliminary empirical study to answer the question. Specially, we gather M-UI-related reviews, and compare the average rating of M-UI-related reviews and total reviews of each app. We observe that the M-UI is concerned by users and the average rating for M-UI-related reviews is lower than the average rating for total reviews. By extracting the topics of M-UI-related reviews, we estimate the sentiment of the M-UI-related topics. The results show that the number of M-UI-related topics are about three or four, and the sentiment of M-UI-related topics is related to the app itself. Further, by investigating the relation between the M-UI-related topics and M-UI design. We observe that users are concerned about the M-UI usability the most, and it is the various aspects of the M-UI that are causing user frustration. In particular, our findings show that M-UI-related reviews reflect the severity of M-UI-related issues and app reviews can help developers to improve M-UI design about appearance, usability, fault-tolerance, of which usability deserves the most attention.

Keywords: Mobile user interface, App reviews, M-UI design

1 Introduction

Mobile user Interface (M-UI) is the medium of information exchange between the system. A good M-UI design makes an app easy, practical, and efficient to use, which significantly affects the success of the app and the loyalty of its users [1-4]. In practice, M-UI-related updates may lead to a higher rate of complaints from users [5]. The quality of M-UI design is greatly affected by subjective factors, developers cannot use a quantitative standard to measure it. With a poorly designed Android GUI, users would feel frustrated and uninstall the application [6]. Different from desktop and software applications, mobile applications have shorter development cycles, M-UI design problems are more prominent. Previous research has been devoted to investigating the rationality of the M-UI. Researchers usually focus on M-UI design [7-8] and testing [9]. For example, Nilsson presented a structured collection of user interface design patterns for mobile applications [7]. Srivastava et al. [8] designed a more user-friendly M-UI for people with low literacy. Alegroth et al. [9] conducted an empirical study to try a more reasonable M-UI layout. Researchers have been looking for ways to enhance M-UI quality for a long time. However, most of the work comes from the perspective of developers. There is still a lack of study on the user perspective about meliorating M-UI design. In this work, we will try to work from the perspective of app reviews. Conducting such research faces serious challenges. First, we all know that there is a lot of noise in app reviews. The research [10] shows that M-UI-related reviews only make up a small percentage of the large number of noise app reviews. Second, the user's perception of the M-UI is valuable but limited to subjective factors. Thus, we plan to bridge the gap between M-UI design and user's perception of the M-UI.

Nowadays, the user's perception of an app can be captured directly from the app market. The app market (e.g., Google Play Store, App Store) provides a platform for users to discuss their experience of using an app. The feedback users leave in the app market is called app reviews. App reviews are judgments made by users after they purchase or experience a product and contain a lot of useful information (e.g., feature requests, annoying bugs) [11]. App review mining has a certain impact on the success of the mobile application, which can contribute to the design of the mobile application [12]. It can also detect erroneous apps through the classification of app reviews, or guide the requirements engineering by extracting bug reports and feature requests

*Corresponding Author: Yong Wang; E-mail: yongwang@ahpu.edu.cn
DOI: 10.53106/160792642023072404013

[13], or app testing and maintenance [14-17], etc. **Therefore, it is meaningful to explore whether app reviews can help developers to improve M-UI design.**

In this paper, we seek to override the challenges mentioned above, and explore whether app reviews can help developers improve M-UI design. In particular, we randomly selected six apps with different categories from Google Play Store and App Store. We get M-UI-related reviews with the help of keywords extraction. Man et al. [18] listed the keywords that are relevant to M-UI in the research. Further, we use word embedding technology [19] to seek words similar to those keywords and filter the result manually. Then, filter M-UI-related reviews in app reviews. In addition, extract the topics of M-UI-reviews, and use sentiment analysis to estimate the M-UI-related topic sentiment. Finally, analyze the relation between the M-UI-related topics and M-UI design manually.

The main contributions of this paper are as follows:

1) We are aware of the severity of the M-UI-related issues by finding the average rating for M-UI-related reviews is lower than the average rating for total reviews.

2) We analyze the relation between M-UI-related topics and M-UI design, and find app reviews can help developers to improve M-UI design about appearance, usability, interaction fault-tolerance, of which usability deserves the most attention.

The rest of the paper is organized as follows: Section 2 introduces the detailed process of our approach. Section 3 presents the research questions and explains the data source, then designs the experiments to answer research questions. Section 4 presents the experimental results. Section 5 lists the potential threats to the validity of this work. Finally, Section 6 concludes the paper.

2 Approach

In this section, we will introduce the framework for this work, which includes four main modules as shown in Figure 1:

1) Data collecting: picking up the M-UI-related reviews from app reviews and tagging them.

2) Pre-processing: removing irrelevant and noisy information from M-UI-related reviews.

3) Topic extraction and analysis: M-UI-related topics are extracted from the M-UI-related reviews, and estimating sentiment of extracted M-UI-related topics.

4) Relation extraction: manually analysis the relation between M-UI-related topics and M-UI design, guided by the sentiment of M-UI-related topics.

In the following, we will explain the process for each of the above modules.

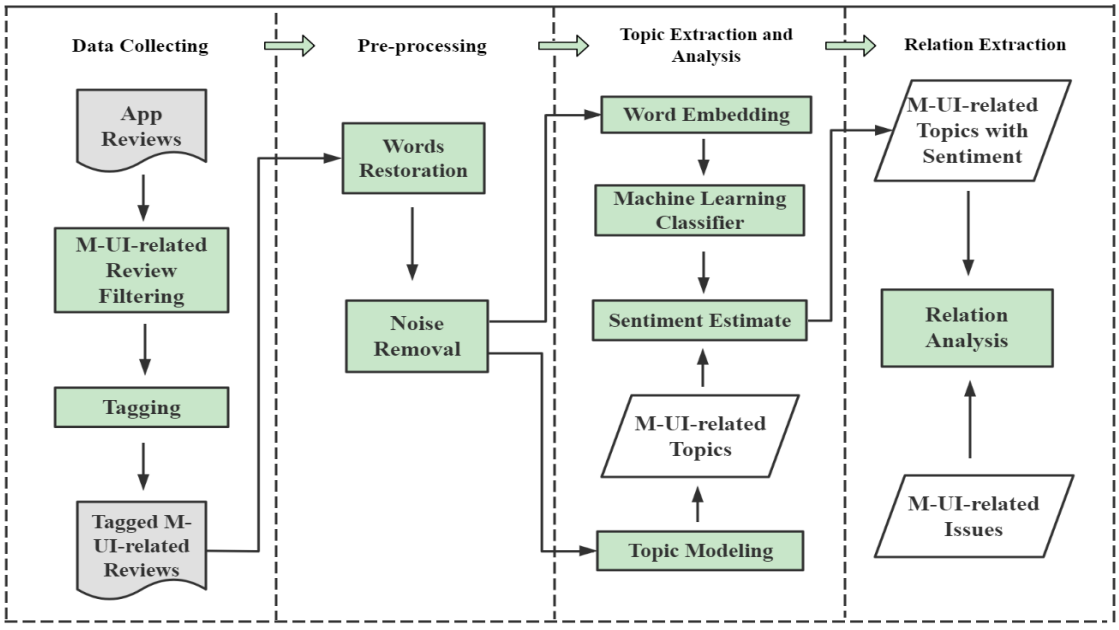


Figure 1. The framework of our approach

2.1 Data Collecting

Since our work focuses on user perceptions of the M-UI, we need to pull them out of app reviews and take those reviews as the dataset.

In order to gather the M-UI-related reviews, we use the keywords extraction method in this work, which is a simple, efficient and widely used way to retrieval. The first thing to do is get the M-UI-related keywords in app reviews. Man et al. [18] studied the complaints of app reviews, they divided the problems into seven categories and listed the keywords (e.g., layout, interface, etc.) related to M-UI. In addition, to get more target reviews, more M-UI-related keywords are needed. Thus, we split every review into words by NLTK (Natural Language Toolkit) [20], and represent words as vectors by Word2Vec [21]. And find out the approximate words are similar to the M-UI-related keywords, then those similar words are chosen manually. The preparation for keywords extraction is ready.

Next, we match each word in every app review from the total reviews with the M-UI-related keywords. If an app review contains at least one M-UI-related keywords, we add that review to the dataset. On the contrary, we discard it. As a result, we get a dataset of M-UI-related reviews. Then, label M-UI-related reviews according to their ratings. For details on labels, see section 3.3.2.

2.2 Pre-processing

The pre-processing step is to remove noisy reviews and special characters from M-UI-related reviews. This is a common step for filtering noisy app reviews [10, 22].

Due to the simplicity and openness of the mobile applications, app reviews are often short and not structured text, which can lead to misspelling, slang, etc. So, it is necessary to reduce the noise in app reviews [23]. First, we use NLTK to break each M-UI-related review into words. Then, in order to correct misspelled words, we use algorithms [24] to replace misspelled words with one of the most likely words from a corpus of common words (e.g., "hella" to "hello"). After that, reduce words to root form by lemmatization (e.g., "playing" to "play"). Finally, remove stop words from reviews. Stop words occur frequently and have little impact on semantics, so we generally remove them as interfering words in text processing. Stop words are provided by NLTK and predefined stop words. Predefined stop words are non-information words manually selected from nearly 1000 app reviews by our team members. The box below lists 10 of 101 non-information words due to space limitation. So far, we have completed noise filtering for M-UI-related reviews.

Predefined stop words: app, omg, cool, fine, four, none, thank, hello, really, plz.

2.3 Topic Extraction and Analysis

To seek what users are talking about on M-UI. In this work, we use topic modeling to extract M-UI-related topics in M-UI-related reviews and estimate the sentiment of extracted M-UI-related topics.

2.3.1 Topic Modeling

A key goal of data analysis is to identify the common characteristics in data, which is usually to explore what is being discussed in documents in text analysis [25]. To accomplish the tasks above, data scientists use a method called topic modelling, which is an unsupervised learning method to cluster the implied semantic structure of the text to identify the document topic [26]. Topic modeling is very suitable for text-type data and is often used for semantic analysis and text mining in natural language processing. In this study, we use a topic modeling method called LDA (Latent Dirichlet Allocation) [27] to identify the topics of M-UI-related reviews.

2.3.2 Sentiment Analysis

Sentiment analysis is a common task in natural language processing [28]. Due to the rapid development of Internet services, more and more services are provided to users (e.g., app, website, etc.). Sentiment analysis aims at discovering users' perceptions on services, so as to provide business advice to service providers and enable them to make better decisions [29]. Therefore, sentiment analysis is a good tool that can analyze users' sentiment through their feedback. Based on the results, developers can improve their apps. In this study, we use Word2Vec word embedding technology and machine learning classifier to estimate the sentiment of the M-UI-related topics.

2.4 Relation Extraction

Nowadays, M-UI design in the process of software development needs to be jointly participated by developers and customers, and there is no established specification for M-UI design, because M-UI development is limited to many subjective factors. The aesthetic of M-UI varies from person to person. Thus, in this work, we decide to analyze the relation between M-UI-related topics and M-UI design manually. We note that all authors major in computer science, it is not a difficult task for them. In the process of analysis, the three authors were involved in the decision. If two people disagree, a third, more experienced author will step in, offer his own opinion and eventually reach a consensus.

3 Experiment Design

In this section, we expound the research questions which attempt to investigate at first. Then, we show the data source and propose the approaches to answer those questions. In order to assist others to replicate our findings, we provide our data source and code at website: https://github.com/yue-stu/work.

3.1 Research Questions

In this study, we set the following three research questions. The first question explores the basics of M-UI-related reviews. The second and third questions explore whether M-UI-related reviews can help developers improve M-UI.

RQ1: Do users care about the M-UI in app reviews?

RQ2: What are the topics in the M-UI-related reviews?

RQ3: Is there a relation between M-UI-related topics and M-UI design?

3.2 Data Source

The App Store and Google Play Store are the two most widely used platforms for downloading applications. We select suitable apps from the above two platforms. First of all, the number of reviews for the selected app needs to exceed 2000. In order to ensure the generalization of the data, both platforms of the selected apps should be included, and come from different app categories. Finally, we randomly select six apps of the top 100, as shown in Table 1. Six apps come from different application categories, including productivity, shopping, and so on. Four apps from Google Play Store and two from the App Store, with a total of 164,031 reviews. For each piece of data, not only the content of app reviews, but also the user's rating of app, version and other features are included.

Table 1. Subject apps

App name	Category	Platform	Number
Swiftkey	productivity	Google Play	21009
Ebay	shopping	Google Play	35483
Clean Master	tools	Google Play	44327
Viber	communication	Google Play	17126
Noaa Radar	wrether	App Store	8368
YouTube	multimedia	App Store	37718

3.3 Evaluation Approaches

In the following, we will introduce approaches for answering the three research questions presented above one by one.

3.3.1 Approach for Answer *RQ1*

The first question focuses on whether users care about the M-UI in app reviews. Previous research [18, 30] has shown that users will express their perceptions about M-UI. For instance, "clunky interface, hard to get back to a channel once you finish watch a video. no landscape mode for the user interface", users complained that M-UI was clunky and had no landscape mode. The users express an opinion about the M-UI. To figure out whether users care about M-UI in app reviews, we need to find out the reviews that users express their perceptions about M-UI.

We use the keywords extraction method mentioned in Section 2.1 to seek M-UI-related reviews. To gather as many M-UI-related reviews as possible. First, the word vector distribution space of the total app reviews is obtained by Word2Vec and calculate the similarity between the word x and the word y. The distance of the two words was judged by calculating the cosine similarity of x and y:

$$\cos(x, y) = \frac{< x, y >}{\|x\|\|y\|}. \quad \textbf{(1)}$$

After obtaining words similar to the keywords, we manually select the appropriate words with a similarity greater than 0.65. The keywords information is shown in Table 2. If there is at least one keyword in the app review, we take the review as M-UI-related. In addition, to explore the popularity and rating situation of M-UI-related reviews, we calculate the percentage of M-UI-related reviews in the total number of reviews, as well as compare the average rating between M-UI-related reviews and total app reviews of each app.

Table 2. Keywords extension

Total keywords	
Previous work [18]	Similar words
ui, interface, design, layout, gui, ux, clunky, redesign, aesthetic, navigation, usability, desing, sleek, appearance, aesthetically, intuitive, minimalistic, ugly, slick, graphic, unintuitive	guideline, homepage, scheme, hideous, awkward, font, gesture

3.3.2 Approach for Answer *RQ2*

What are the topics that users talk about when discussing the M-UI? In this study, we use LDA to extract M-UI-related topics from the M-UI-related reviews. Then, use sentiment analysis model to estimate the sentiment of M-UI-related topics.

Topic modeling is often used in text analysis to extract document topics. Before topic modeling, the number of topics to extract should be determined. However, the computational perplexity [27] shows that the optimal number of topics is always greater than ten, which is not in line with our expectations. Then, we decide to use pyLDAvis, which is a topic visualization tool. As shown in Figure 2, a circle represents a topic. The distance between the circles represents the connection between the topics. The size of the circle represents the importance of the topic, and the larger the circle, the more important the topic is. We aim to find the maximum number of topics where all the circles do not intersect. After determining the appropriate number of topics, we extract a certain number of topics from M-UI-related reviews. The extracted topic is related to the mobile user interface. Then, establish the sentiment analysis model. We use the Word2vec word embedding tool to get the word vector of the preprocessed M-UI-related reviews, and label them according to their ratings. Studies have pointed out that there is a direct relationship between rating and user sentiment [31], we consider rating less than or equal to three as a negative review, and rating above three as a positive review. We used machine learning classifiers, including Random Forest, SVM (Support Vector Machine), Naive Bayes and Logistic Regression, to predict sentiment in M-UI-related topics. It is noted that the number of positive and negative reviews differed greatly in different app datasets. Thus, we balance positive and negative samples to solve the problem of data imbalance by undersampling. For each app, we compare the performance of each classifier. In order to improve the performance of the classifier, we use grid search for performance tuning. The best classifier is determined by calculating Precision, Recall and F-measure.

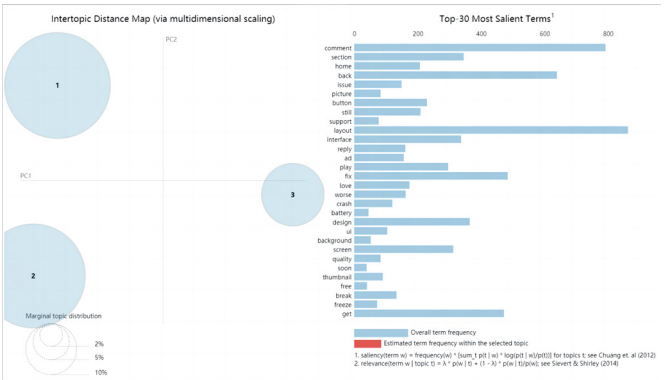


Figure 2. Inter-topic distance map

$$Precision = \frac{TP}{TP + FP}. \quad \textbf{(2)}$$

$$Recall = \frac{TP}{TP + FN}. \quad (3)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (4)$$

Where TP, FP, FN indicate the true positive, false positive and false negatives of the confusion matrix (as shown in Table 3). Finally, input the extracted M-UI-related topics into the trained sentiment analysis model to get the sentiment of each M-UI-related topic.

Table 3. The confusion matrix

Positive		Actual value	
		Negative	
Predicted value	Positive	TP	FP
	Negative	FN	TN

3.3.3 Approach for Answer *RQ3*

The third question concerns whether there is a relation between M-UI-related topics and M-UI design? Since there is no clear standard, we decide to use manual analysis.

We adopt the method of group discussion to make the experiment more rigorous. It is noted that all authors major in computer science and have experience in app development. The work can be divided into two parts. First, the previous work of Nielsen [32], he identified ten principles of UI design. The same applies to the M-UI design. As shown in Figure 3, we map ten principles into three categories of M-UI-related issues with the help of two software engineering professors. Three categories of M-UI-related issues were identified, namely appearance, usability, and fault-tolerance. By manually analyzing the relation between each M-UI-related topic and the above three categories, it can determine which one or more categories this M-UI-related topic belongs to. In the discussion of defining categories, if two authors disagree, a third, more experienced author will step in, offer his opinion, and finally reach a consensus. Second, by analyzing which categories of M-UI problems these topics fall into. we can get a rough idea of what the user perception about the M-UI with the help of sentiment of M-UI-related topic. Through the result, we can infer whether app reviews really help developers improve M-UI design.

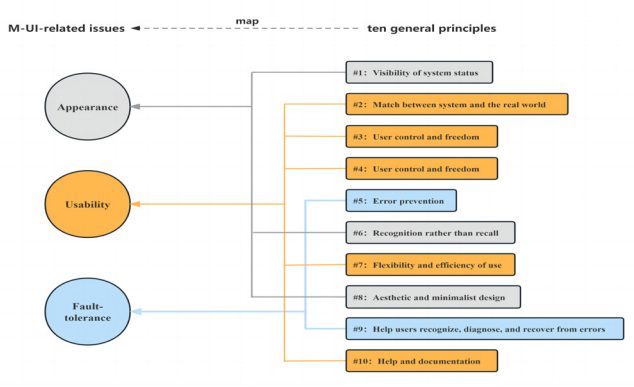


Figure 3. The map of ten principles to M-UI-related issues

4 Results

In this section, we will show the results for answering the questions presented in Section 3.1.

4.1 *RQ1: Do Users Care about the M-UI in App Reviews?*

The first research question is whether users care about the M-UI in app reviews. Previous research got the M-UI-related keywords from app reviews [18]. We use Word2vec to find similar words for these keywords as extensions, then find M-UI-related reviews through keywords extraction. After finding out the M-UI-related reviews, we conduct a statistical analysis. First, we calculate the proportion of the number of M-UI-related reviews to the total app reviews, and compare the average rating of M-UI-related reviews with the average rating of total app reviews. Table 4 shows the result, the proportion of M-UI-related reviews varies greatly among different apps, with the lowest being 0.85% and the highest being 5.17%. It's different from app to app. On the whole, M-UI-related reviews account for about 2.86% of the total reviews. As can be seen from comparison of the last two columns of the table, the average rating of the total review of each app is higher than the average rating of the M-UI-related reviews of each app. The difference between the average rating of total reviews and the average rating of M-UI-related reviews is quite different of each app. Therefore, it is found that the user's evaluation of M-UI is lower than the average, which reflects the severity of the M-UI-related issues. If we figure out where the M-UI is going to improve, rating goes up, the app will be more competitive in the market.

Answer to RQ1: Users care about the M-UI in app reviews. By the statistics, the average rating for M-UI-related reviews is lower than the average rating for total reviews, which reflects the severity of the M-UI-related issues.

4.2 *RQ2: What are the Topics in the M-UI-related Reviews?*

The second question focuses on what topics exist in M-UI-related reviews? We extract the M-UI-related topics by topic modeling and establish a sentiment analysis model to estimate the M-UI-related topic sentiment, and use pyLDAvis to determine the number of M-UI-related topics. Precision, recall, and F-measure are used to measure the quality of sentiment analysis models. Table 5 shows a snapshot of the top seven terms of the M-UI-related topic on YouTube. The words in the topic can reflect the meaning of the topic. For example, the words "picture", "interface", and "design" in topic 1 reflect the appearance and design of M-UI. The performance of the sentiment analysis model is shown in Table 6. From the value of F1-score and accuracy, the Random Forest is better than the other three classifiers in most applications. However, for Viber, SVM performs best. According to the performance of the classifier on each app, the classifier with the best performance is selected as the sentiment analysis model. Next, the established sentiment analysis model is used to analyze the M-UI-related topic sentiment. The results are shown in Table 7. The total number of M-UI-related topics extracted is twenty-two, about three or four for each app. We note that the sentiment of the M-UI-related topic is related to the app itself, and the app with

higher rating has more positive M-UI-related topics than apps with low rating. However, there is not a linear relation between the lower the rating and the number of negative M-UI-related topics.**Answer to RQ2:** Each app has about three or four M-UI-related topics, and the number of positive and negative M-UI-related topics varies greatly depending on the app itself.

Table 4. The statistical results

App name	Total reviews	M-UI-related reviews	Proportion	Average rating of total reviews	Average rating of M-UI-related reviews
Swiftkey	44327	377	0.85%	4.517	4.382
Ebay	35483	1044	2.94%	2.830	2.771
Clean master	21009	1088	5.17%	4.152	4.147
Viber	17126	197	1.15%	3.326	3.102
Noaa radar	8368	346	4.13%	4.454	4.092
YouTube	377118	1642	4.35%	2.175	1877
Total	164031	4694	2.86%		

Table 5. Top seven terms for each topic on YouTube

Topic	Topic 1	Topic 2	Topic 3
Term	still	comment	back
	issue	layout	like
	play	fix	layout
	interface	change	use
	picture	section	watch
	design	make	go
	get	get	screen

Table 6. Performance of different classifiers

App name	Classifier name	Precision	Recall	F1-score	Accuracy
Swiftkey	**Random forest**	0.94	0.89	0.91	0.89
	SVM	0.76	0.72	0.67	0.72
	Naive bayes	0.70	0.74	0.69	0.74
	Logistic regression	0.71	0.71	0.65	0.71
Ebay	**Random forest**	0.85	0.82	0.83	0.82
	SVM	0.56	0.53	0.52	0.53
	Naive bayes	0.72	0.70	0.71	0.70
	Logistic regression	0.56	0.54	0.53	0.54
Clean master	**Random forest**	0.95	0.79	0.85	0.79
	SVM	0.58	0.54	0.51	0.54
	Naive bayes	0.58	0.57	0.54	0.57
	Logistic regression	0.58	0.57	0.54	0.57
Viber	Random forest	0.71	0.68	0.68	0.68
	SVM	0.84	0.78	0.78	0.78
	Naive bayes	0.69	0.68	0.68	0.68
	Logistic regression	0.79	0.72	0.73	0.73
Noaa radar	**Random forest**	0.89	0.81	0.84	0.81
	SVM	0.74	0.51	0.48	0.51
	Naive bayes	0.74	0.57	0.53	0.67
	Logistic regression	0.62	0.57	0.52	0.57
YouTube	**Random forest**	0.97	0.84	0.90	0.84
	SVM	0.73	0.74	0.74	0.74
	Naive bayes	0.70	0.72	0.71	0.72
	Logistic regression	0.55	0.51	0.45	0.51

Table 7. M-UI-related topic number and M-UI-related topic sentiment

App name	M-UI-related topic number	Positive	Negative
Swiftkey	1	√	
	2	√	
	3	√	
	4	√	
Ebay	1		√
	2	√	
	3	√	
	4		√
Clean master	1	√	
	2	√	
	3	√	
Viber	1		√
	2	√	
	3		√
	4		√
Noaa radar	1	√	
	2	√	
	3	√	
	4	√	
YouTube	1	√	
	2		√
	3		√

4.3 *RQ3: Is There a Relation between M-UI-related Topics and M-UI Design?*

The last question concerns whether there is a relation between M-UI-related topics and M-UI design? We classify the M-UI-related topics to the M-UI-related issues one by one manually, and the results are shown in Table 8. Among the three categories (appearance, usability and fault-tolerance), we find that most M-UI-related topics are related to usability and appearance, while there are few M-UI-related topics about fault-tolerance. In addition, most M-UI-related topics with a positive sentiment are related to usability, while those with a negative sentiment are involved all categories. It can be concluded that the user cares most about the usability, while appearance is secondary. In the case of negative M-UI-related topics, we suggest that the experience should be prioritized, while the appearance should also be paid attention to. We also note that M-UI-topics with negative sentiment include other situations, which includes M-UI changes. Therefore, we speculate that frequent changes to the M-UI will also lead to user dissatisfaction.

Answer to RQ3: There is a relation between M-UI-related topics and M-UI design. Users care about the appearance, usability and fault-tolerance of M-UI, among which the usability is the most prominent.

Table 8. Correspondence between M-UI-related topics and M-UI-related issues

App name	M-UI-related topic number	Appearance	Usability	Fault-tolerance
Swiftkey	1	√	√	
	2		√	
	3		√	
	4		√	
Ebay	1		√	
	2	√		
	3		√	
	4	√	√	
Clean master	1		√	
	2		√	
	3	√	√	
Viber	1	√		
	2		√	
	3		√	
	4	√		
Noaa radar	1	√	√	
	2	√	√	
	3	√	√	
	4		√	
YouTube	1	√	√	
	2		√	√
	3	√	√	

5 Threats to Validity

In this section, we discuss the threats to validity of our work.

Internal Validity. In terms of internal threats, the results of this experiment may be influenced by manual analysis (including selecting similar keywords and judging M-UI-related topic categories, etc.). To reduce the impact of this risk on the results, we assign a complete operational process (including examples and standards) as experimental guidance. It is worth noting that all the experimental personnel majored in computer science and have practical experience in software development. In addition, stop words are provided by NLTK and predefined stop words. Predefined stop words are chosen by our team manually. The unanimous agreement of team members is required when choosing predefined stop words. We randomly selected 200 pieces of review from the dataset and the false-positive rate was about 2.5%.

External Validity. This is related to factories from an external aspect. Our results may have been skewed by the data source, we search just about 160,000 reviews, which is nowhere near the number of reviews in the entire app market. Therefore, to enhance the diversity of the data, we randomly selected six of the top 100 apps from the most popular app download platforms (Google Play Store and App Store). What's more, six different apps from different app categories. Different software markets represent different user behaviors and characteristics. Similarly, different types of app users are different.

6 Conclusion

In this paper, we seek whether app reviews could be used to improve M-UI design. We randomly selected apps in six different categories from the Google Play Store and app Store, with over 160,000 app reviews. Firstly, we find out the M-UI-related reviews from the app reviews, and compare the average rating of M-UI-related reviews and total app reviews of each app. Then, extract the M-UI-related topics in the M-UI-related reviews, and estimate the sentiment of the M-UI-related topics. Finally, analyze the relation between the M-UI-related topics and M-UI design through manual analysis. From the experiment, we find that M-UI is concerned by users, and the average rating for M-UI-related reviews is lower than the average rating for total app reviews, which reflects the severity of M-UI-related issues. There are about three or four M-UI-related topics that users talk about M-UI, and the M-UI-related topic sentiment is related to the app itself. Users care about the usability the most, and users complain about the M-UI for a variety of reasons. The results of our study are helpful for developers to improve M-UI design.

In the future, we will expand our work vertically and horizontally, and hope to accomplish more fine-grained extraction of M-UI-related reviews, as well as more rational extraction of topics. In addition, we are exploring more ways to improve the quality of M-UI.

Acknowledgment

This work was supported by the Key Project of Anhui University Natural Science Foundation (Grant No. YJS20210453, KJ2020A0361, KJ2021A1028), Anhui Province scientific research planning project(Grant No. 2022AH050953), National Natural Science Foundation of China under project (Grant No. 62002084, 61976005), the University Synergy Innovation Program of Anhui Province (Grant No. GXXT-2022-047), the Key Project of Natural Science Research of Higher Education Institution of Anhui Province of China (Grant No. KJ2020A0363, KJ2021A1028), Natural Science Foundation of Zhejiang Province (Grant No.LQ21F020004), Stable support plan for colleges and universities in Shenzhen (Grant No. GXWD20201230155427003-20200730101839009).

References

[1] B. J. Jansen, The graphical user interface, *ACM SIGCHI Bulletin*, Vol. 30, No. 2, pp. 22-26, April, 1998.

[2] H. Khalid, E. Shihab, M. Nagappan, A. E. Hassan, What Do Mobile App Users Complain About? *IEEE Software*, Vol. 32, No. 3, pp. 70-77, May-June, 2015.

[3] C. Y. Huang, M. C. Yang, C. Y. Huang, An Empirical Study on Factors Influencing Consumer Adoption Intention of an AI-powered Chatbot for Health and Weight Management, *International Journal of Performability Engineering*, Vol. 17, No. 5, pp. 422-432, May, 2021.

[4] D. Li, W. E. Wong, M. Chau, S. Pan, L. S. Koh, A Survey of NFC Mobile Payment: Challenges and Solutions using Blockchain and Cryptocurrencies, *2020 7th International Conference on Dependable Systems and Their Applications (DSA)*, Xi'an, China, 2020, pp. 69-77.

[5] S. Hassan, C. Bezemer, A. E. Hassan, Studying bad updates of top free-to-download apps in the google play store, *IEEE Transactions on Software Engineering*, Vol. 46, No. 7, pp. 773-793, July, 2020.

[6] B. Yang, Z. Xing, X. Xia, C. Chen, D. Ye, S. Li, Uis-hunter: Detecting UI design smells in android apps, *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, Madrid, Spain, 2020, pp. 89-92.

[7] E. G. Nilsson, Design patterns for user interface for mobile applications, *Advances in Engineering Software*, Vol. 40, No. 12, pp. 1318-1328, December, 2009.

[8] A. Srivastava, S. Kapania, A. Tuli, P. Singh, Actionable UI design guidelines for smartphone applications inclusive of low-literate users, *Proceeding of the ACM on Human-Computer Interaction*, Vol. 5, Article No. 136, pp. 1-30, April, 2021.

[9] E. Alegroth, Z. Gao, R. A. P. de Oliveira, A. M. Memon, Conceptualization and evaluation of component-based testing unified with visual GUI testing: An empirical study, *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, Graz, Austria, 2015, pp. 1-10.

[10] Q. Chen, C. Chen, S. Hassan, Z. Xing, X. Xia, A. E. Hassan, How should I improve the UI of my app? A study of user reviews of popular apps in the google play, *ACM Transactions on Software Engineering and Methodology*, Vol. 30, No. 3, pp. 37:1-37:38, July, 2021.

[11] C. Gao, J. Zeng, F. Sarro, D. Lo, I. King, M. R. Lyu, Do users care about ad's performance costs? exploring the effects of the performance costs of in-app ads on user experience, *Information and Software Technology*, Vol. 132, Article No. 106471, April, 2021.

[12] X. Gu, S. Kim, what parts of your apps are loved by users?, *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Lincoln, NE, USA, 2015, pp. 760-770.

[13] W. J. Martin, F. Sarro, Y. Jia, Y. Zhang, M. Harman, A survey of app store analysis for software engineering, *IEEE Transactions on Software Engineering*, Vol. 43, No. 9, pp. 817-847, September, 2017.

[14] W. Maalej, H. Nabil, Bug report, feature request, or simply praise? on automatically classifying app reviews, *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, Ottawa, Canada, 2015, pp. 116-125.

[15] R. Luo, S. Huang, H. Chen, M. Y. Chen, Code Confusion in White Box Crowdsourced Software Testing, *International Journal of Performability Engineering*, Vol. 17, No. 3, pp. 276-288, March, 2021.

[16] G. Grano, A. Ciurumelea, S. Panichella, F. Palomba, H. C. Gall, Exploring the integration of user feedback in automated testing of android applications, *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Campobasso, Italy, 2018, pp. 72-83.

[17] Y. Qu, W. E. Wong, D. Li, Empirical Research for Self-admitted Technical Debt Detection in Blockchain Software Projects, *International Journal of Performability Engineering*, Vol. 18, No. 3, pp. 149-157, March, 2022.

[18] Y. Man, C. Gao, M. R. Lyu, J. Jiang, Experience report: Understanding cross-platform app issues from user reviews, *2016 IEEE 27th International Symposium on Software Reliability Engineering*, Ottawa, Canada, 2016, pp. 138-149.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, Nevada, United States, 2013, pp. 3111-3119.

[20] S. Bird, E. Loper, NLTK: the natural language toolkit, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 2004, pp. 214-217.

[21] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *1st International Conference on Learning Representations*, Scottsdale, Arizona, USA, 2013, pp. 1-12.

[22] E. Guzman, W. Maalej, How do users like this feature? A fine grained sentiment analysis of app reviews, *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, Sweden, 2014, pp. 153-162.

[23] N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, B. Zhang, Ar-miner: mining informative reviews for developers from mobile app marketplace, *Proceedings of the 36th International Conference on Software Engineering*, Hyderabad, India, 2014, pp. 767-778.

[24] Debugfanfan, English word error correction, *https://github.com/debugfanfan/EnglishWordErrorCorrection/*, May, 2020.

[25] I. Vayansky, S. A. P. Kumar, A review of topic modeling methods, *Information system*, Vol. 94, Article No. 101582, December, 2020.

[26] D. M. Blei, Probabilistic topic models, *Communications of the ACM*, Vol. 55, No. 4, pp. 77-84, April, 2012.

[27] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research*, Vol. 3, pp. 993-1022, January, 2003.

[28] I. Chaturvedi, E. Cambria, R. E. Welsch, F. Herrera, Distinguishing between facts and opinions for sentiment analysis: Survey and challenges, *Information Fusion*, Vol. 44, pp. 65-77, November, 2018.

[29] M. Birjali, M. Kasri, A. B. Hssane, A comprehensive survey on sentiment analysis: Approaches, challenges and trends, *Knowledge-Based Systems*, Vol. 226, Article No. 107134, August, 2021.

[30] E. Noei, F. Zhang, Y. Zou, Too many user-reviews! what should app developers look at first? *IEEE Transactions on Software Engineering*, Vol. 47, No. 2, pp. 367-378, February, 2021.

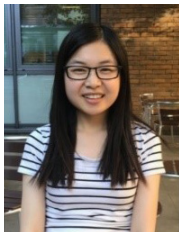
[31] L. Hoon, R. Vasa, G. Y. Martino, J. Schneider, K. Mouzakis, Awesome!: conveying satisfaction on the app store, *OzCHI' 13: Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation*, Adelaide, Australia, 2013, pp. 229-232.

[32] J. Nielsen, Enhancing the explanatory power of usability heuristics, *Conference on Human Factors in Computing Systems*, Boston, Massachusetts, USA, 1994, pp. 152-158.

Biographies

Wenge Le received the B.S. degrees in automation from Huangshan university. He is a graduate student at Anhui Polytechnic University, China. His current research interests include software testing, fault localization, and program debugging.

Yong Wang received his B.S. and M.S. degrees in computer science from Anhui Polytechnic University, and he received his Ph.D. degree in computer science and technology from Nanjing University of Aeronautics and Astronautics. His current research interests include software testing, fault localization, and program debugging.



Cuiyun Gao received the B.S. degree from the department of Communication Engineering, Shanghai University, and the Ph.D. degree from the department of computer science and engineering, Chinese University of Hong Kong. Her research interests include software repository mining and natural language processing.

Liangfen Wei received her M.S. degrees in computer science from Hefei Polytechnic University. Her current research interests include machine learning systems and Software defect prediction.

Fei Yang received his B.S. and M.S. degrees in computer science from Shanghai Jiao Tong University, and he received his Ph.D. degree in computer science from Eindhoven University of Technology. His current research interests include deep learning, machine learning systems, and concurrency theory.