information

MDPI

Article

A Sentiment-Statistical Approach for Identifying Problematic Mobile App Updates Based on User Reviews

Xiaozhou Li *, Boyang Zhang *, Zheying Zhang * and Kostas Stefanidis *

Faculty of Information Technology and Communication Sciences (ITC), Tampere University,
33520 Tampere, Finland

* Correspondence: xiaozhou.li@tuni.fi (X.L.); boyang.zhang@tuni.fi (B.Z.); zheying.zhang@tuni.fi (Z.Z.); konstantinos.stefanidis@tuni.fi (K.S.)

Received: 22 January 2020; Accepted: 10 March 2020; Published: 12 March 2020

Abstract: Mobile applications (apps) on IOS and Android devices are mostly maintained and updated via Apple Appstore and Google Play, respectively, where the users are allowed to provide reviews regarding their satisfaction towards particular apps. Despite the importance of user reviews towards mobile app maintenance and evolution, it is time-consuming and ineffective to dissect each individual negative review. In addition, due to the different app update strategies, it is uncertain that each update can be accepted well by the users. This study aims to provide an approach to detect the particular days during the mobile app maintenance phase when the negative reviews require developers' attention. Furthermore, the method shall facilitate the mapping of the identified abnormal days towards the updates that result in such negativity in reviews. The method's purpose is to enable app developers to respond swiftly to significant flaws reflected by user reviews in order to prevent user churns.

Keywords: mobile app; sentiment analysis; maintenance; update; user review; exponential power distribution; Word2Vec

1. Introduction

Contemporary mobile applications (apps) play an increasingly important role in people's daily lives, which are directly influenced by the apps' quality. Via the dynamic way of distribution supported by the platforms e.g., Apple Appstore and Google Play, the release periods of mobile apps are largely shortened compared to those of traditional software products [1]. Due to the fiercely competitive mobile app market, the developers are obliged to constantly update their app products by fixing bugs, improving interfaces, adapting to system updates, and providing new features, in order to maintain the quality of the apps and to increase user retention and satisfaction [1,2]. In the constant updating process, understanding end users' needs and requests is important and requires enormous effort. There are various ways of getting access to users' feedback for a particular app. Besides the dedicated in-app user feedback collecting tools, the online app stores contain a mechanism to allow users to provide text reviews and rating scores, which enhances their importance as a stakeholder [3].

The reviews given by end users are helpful towards improving the app quality in general through its maintenance and evolution with proper analysis, despite the proportion of informative reviews is around one third [4,5]. Many studies have proposed approaches and techniques facilitating the analysis of mobile app user reviews [5–11]. Therein, sentiment analysis and topic modeling are often applied for the analysis with various focuses, e.g., on detecting review characteristics [12], on identifying review inconsistency and user concerns [11], on review classification [7,8], on the controversiality about the sentiment towards specific entities [13], or even on comparing them with users ratings [14].

Furthermore, many studies also contribute to applying other methods of review analysis in order to tackle other software maintenance and evolution related issues, e.g., release planning [15,16], change requests localizing and recommendation [17], as well as software suggestions [18].

On the other hand, due to the dynamic distribution mechanism and constant market changes, mobile apps are continuously updated with a rapid pace, though such an update mechanism implementation is risky for potential user dissatisfaction [19]. Despite most users being happy with the apps with frequent updates but hesitate to install them, worrying about potential hazards [1]. Most of the previous studies focus on eliciting users' general opinion from the reviews regarding a particular mobile app as a whole, while also lacking support on the detection of updates, which may adversely affect user experience and satisfaction. We call such updates problematic updates. Releasing continuously problematic updates can stop users from using the particular app again, which in turn impacts the app's exposure opportunities on the app stores. Identifying the problematic updates early on and understanding the causes of user dissatisfaction can help mobile app providers predict potential problems before releasing a new update which ensures the success of the app in its life cycle. Li et al. propose an approach to analyzing the topic and sentiment changes before and after a particular update, but has a lack of support for identifying the particular time and the potential update that requires attention [10]. Xia et al. provide a way to predict mobile app crashing updates based on commit data analysis, but lack of facilitation towards identifying generally problematic updates [20].

In this paper, we propose a sentiment-statistical approach of identifying the problematic mobile app updates based on user review analysis, by specifically answering the following research questions:

1. How to identify the collective dissatisfaction of users based on their reviews?
2. How to verify it is the recent update that results in the users' dissatisfaction?

The reminder of the article is organized as follows. Section 2 introduces the related studies regarding similar topics. Section 3 presents our method with details. Section 4 presents a case study, applying the proposed method on retrieved mobile app review data. Section 5 further discusses the relevant issues, as well as the limitation of the study and future works. Section 6 concludes the article with a summary of our contributions.

2. Related Work

Regarding the use of data mining techniques for user reviews analysis towards mobile app quality, as well as practice regarding maintenance and evolution, many previous studies focus on aspects, like the helpfulness and informativeness of the reviews, feature extraction and review classification [21]. For example, towards informative review identification, Chen et al. adopt the expectation maximization for the naive Bayes method to classify informative and non-informative reviews and topic modeling methods to group informative reviews [5]. Gao et al. adopt the Info-rate index to analyze the informative rate of the reviews and track the dynamics of top-rated reviews without manual labeling [22]. Chandy and Gu propose a method to identify spam reviews with the baseline decision tree model and latent class model [23]. Towards feature extraction, Vu et al. propose a keyword-based framework for semi-automated review analysis facilitating the extraction of review keywords and the mapping to the related negative reviews [24]. Fu et al. adopt statistical analysis and topic modeling method to discover inconsistency in reviews and identify the major concerns of the users marked by extracted keywords [11]. Many other studies also use topic modeling methods, such as, LDA and ASUM, to extract major concerns of users from the reviews and the related features [4,6,25]. Furthermore, despite qualitative and exploratory methods also being used for the classification of user reviews, specifically the complaint types of the users [2,26], many scholars still adopt natural language processing, topic modeling and sentiment analysis techniques for such purpose [8,10,27].

Together with the opinion mining of user reviews, many studies focus on the continuous maintenance of mobile apps via update analysis and release planning. Villarroel et al., propose the

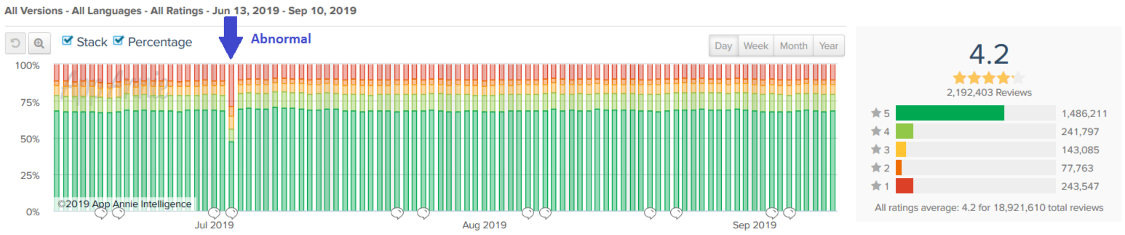
CLAP method to categorize and cluster user reviews based on extracted features and to prioritize and recommend review clusters towards the planning of the subsequent updates [15]. Ciurumelea et al., propose the user request referencer prototype to not only classify reviews but also map the reviews according to source code files that can be modified to address the issues within [16]. On the other hand, towards the analysis of mobile app updates, Wang et al. use a k-means clustering algorithm to identify seven mobile app update patterns based on the feature intensity trend between two neighboring updates, reflecting the common update behaviors towards acting on user reviews [28]. Li et al, present a method with topic modeling and sentiment analysis to analyze the changes of user opinions through continuous app updates [29]. Overall, the previous works on mobile app reviews opinion mining largely focus on issues related to detection, and provide corresponding strategies for future updates. This study, on the other hand, aims to facilitate the identification of particular updates that result in statistically abnormal amount of negative reviews.

3. Method

In this work, we propose an approach aiming to identify the periods of time when the user reviews of a particular mobile app reflect noticeable amount of negativity. Such identification of the opinion changes shall also be correlated with the changes of the topics in the according review periods. This approach also aims to identify the particular update that is most likely connected to such sentiment and main topic changes by comparing the similarity between the keywords of the update content and those of the reviews from the period of such changes. In general, this approach encompasses two key steps: (1) identifying the abnormal days of changes in the sentiment of user reviews, and (2) identifying the corresponding problematic update that is most likely connected to such abnormality. The outcomes of these two steps, respectively, answer the research questions mentioned above.

3.1. Sentiment Change Distribution

In order to identify the period of time during the maintenance and evolution of a particular mobile app, where the overall user review sentiment of that period is abnormal, we shall firstly be able to differentiate such abnormality from normal review sentiment. By observing the rating percentage of changes of a number of popular mobile apps from the last 90 days (from 13 June 2020 to 10 September 2020), we find that the rating percentages are largely stable, despite the varied proportion number from app to app. For example, such stability in the rating proportions through days can be easily observed in Figure 1, which shows the rating percentage changes for 90 days for mobile app Whatsapp (Obtained from https://www.appannie.com).


All Versions - All Languages - All Ratings - Jun 13, 2019 - Sep 10, 2019

☑ Stack ☑ Percentage

Abnormal

Day Week Month Year

4.2

2,192,403 Reviews

★ 5	1,486,211
★ 4	241,797
★ 3	143,085
★ 2	77,763
★ 1	243,547

All ratings average: 4.2 for 18,921,610 total reviews

©2019 App Annie Intelligence

Figure 1. The rating percentage changes of Whatsapp from 13 June 2020 to 10 September 2020.

Figure 1 shows an abnormal rating proportion that is noticed on the 3rd July 2019, when the percentage of 1-star ratings rose from an average 10% up to 28.4%, while that of 5-star ratings dropped from 70% down to 47.4%. Such sudden rating proportion changes on that particular day can be assumed to be reflecting the changes in the quality of the app via the recent update. On the other hand, user ratings are somehow inconsistent towards the review content provided [11], when the reviews are considered more accurately reflecting the user's opinion. In addition, based on the results from the 1-year data of reviews and ratings from the five mobile apps given next (3,793,125 reviews), we find the correlation between the daily average sentiment and ratings is high (with Pearson's $r = 0.984$). Due to such high correlation between ratings and review sentiment, the phenomenon of sudden changes in rating proportion caused by updates can also be observed by the changes in the collective review sentiment.

Accordingly, when considering such phenomena of daily average review sentiment being stable, we can then propose a hypothesis that having the review data divided by a fixed time period (e.g., by day), the changes of the average sentiments of the obtained review divisions along the timeline are normally distributed. To test the hypothesis, we use the Kolmogorov–Smirnov test (K-S test) [30] for the fitness of negative sentiment changes of the previously mentioned data of five mobile apps to normal distribution. However, the hypothesis does not stand with p taking value equal to 0.000001 (less than 0.05). Furthermore, in order to find the best fitting distribution model, we apply the K-S test for 86 other different distribution models (https://docs.scipy.org/doc/scipy/reference/stats.html) finding that our data fits best to generalized normal distribution (version 1. Exponential power distribution, shown in Figure 2) [31] with a p value of 0.968 ($\mu = 0.0002$, $\alpha = 0.0227$, $\beta = 1.0444$).

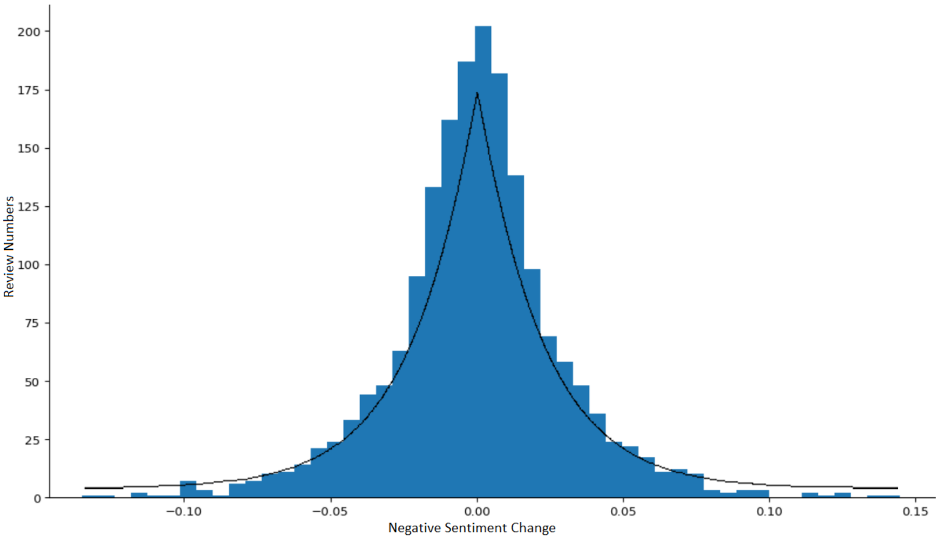


Figure 2. The distribution of the review data negative sentiment change.

3.2. *Identify Abnormal Sentiment Changes*

The overall review sentiment of a particular day can be defined in various ways, such as the average sentiment score of the reviews on each day or the rate of certain type of sentiment (i.e., positive or negative). For review sentiment analysis, due to the inconsistency between the sentiment expressed by users and the actual quality, it is highly likely the majority of moderate negative reviews can be neutralized by the minority positive reviews with exaggerated expression. Furthermore, as the method is to identify the abnormal day when a significant number of negative reviews occur, we hereby define the sentiment change between two days as the increase or decrease rate of negative review numbers, where the strength of sentiment is chosen to be ignored. Thus, considering respectively n_i and n_{i+1} negative reviews found in the N_i reviews on the day d_i and N_{i+1} reviews on the next day d_{i+1}, the review sentiment changes between day d_i and day d_{i+1}, denoted as as_i, is calculated as $as_i = (n_{i+1}/N_{i+1}) - (n_i/N_i)$.

Based on the hypothesis proposed previously, if the variables of daily sentiment changes fit normal distribution, we can then consider the samples that lie within the band around the mean in a normal distribution with a width of six standard deviations as normal (i.e., the three sigma rule [32]). However, for the non-normal distributions, such inference towards the percentage of normal values may vary. Based on the previous exponential power distribution model for the review sentiment change data, we simulate 1,000,000 samples having such distribution with the obtained parameters (i.e., $\mu = 0.0002$, $\alpha = 0.0227$, $\beta = 1.0444$). The simulation is from the R package *gnorm* (https://cran.r-project.org/web/packages/gnorm/index.html). The calculated number of samples lying in the band around the mean with $\pm 3\sigma$ width is 986,424, indicating 98.6% of the values shall be considered normal. Thus, the abnormal sentiment changes of the reviews are the ones lying outside the band. On the other hand, considering Chebyshev's inequality [33], no more than $1/k^2$ of the distribution's values can be more than k standard deviations away from the mean. Specifically, regarding the simulation data, the abnormal values are the $1 - 1/k^2 = 0.986$ of the distribution, that is, around $k = 8$ standard deviations away from the mean. Therefore, regarding our dataset, it is reasonable to consider the days when the sentiment change values lying out the 98.6% band around the mean as abnormal.

3.3. Identify Problematic Updates

Despite the changes in review, sentiment scores can be used to identify the potential problems in the mobile app quality, it is possible that such changes result from the general quality deterioration instead of from particular problematic updates [34]. A way of finding the connection between the identified abnormal review sentiment changes and the corresponding problematic updates is to detect the similarity between the content of the reviews that causing the changes and that of the updates. Towards such purpose, we adopt the *Word2Vec* tool [35].

The Word2Vec tool uses vectors to represent words with efficient and continuous bag-of-words and skip-gram schemes [36]. Each word from the text corpus generated from the text data is transformed into an individual vector. The vocabulary model is constructed by training the text data, which uses a log-linear classifier to predict words occurring within a certain range to either side of the word and learn the word representation. Simply put, for a particular word in the corpus, it is less likely related to the words occurring frequently far away from it. Thus, the similarity between this word and the words far away weighs less.

In order to investigate the similarity between the content of reviews and that of the update, we firstly train the Word2Vec model with the textual review data of a particular app. After identifying the days when the sentiment changes are considered abnormal, we summarize the negative reviews of that day using the term frequency–inverse document frequency (TF-IDF), which reflects the importance of a word to a document (i.e., review text) in a collection of corpus (i.e, collection of reviews) [37]. Thereafter, the similarity between the content of a particular update and that of the negative reviews of the identified abnormal days can be measured by averaging the similarities between the words with high TF-IDF value of the abnormal-day negative reviews and the keywords extracted from the update description text. Ideally, when the main cause of a particular abnormal day is verified, it is most likely the nearest update before the abnormal day that results in such abnormality. Otherwise, further investigation into the details of the reviews is required to verify the causes.

3.4. Algorithm

Algorithm 1 offers an overview of the proposed approach. Specifically, our approach consists of three main steps:

1. Calculate the parameters of the exponential power distribution model of the daily review sentiment changes.
2. Identify the abnormal sentiment change days based on the distribution model, if any.
3. Check whether it is the nearest previous update that is problematic and causing such detected abnormality in user review sentiment change by comparing the similarity between the negative reviews of the abnormal days and the update description texts.

Algorithm 1: Algorithm of identifying abnormal days and problematic updates.

```
Data: Set of Reviews within a Fixed Period
Result: Identified Problematic Update if Abnormal Sentiment Change Detected
INITIALIZATION;
$R \leftarrow$ set of reviews;
for each $r_i \in R$ do
    $s_i$ ($\in S$) $\leftarrow$ getSentimentScore($r_i$)
end
$D \leftarrow$ set of days, where R is obtained;
for each {$r_x$, $r_{x+1}$, ... $r_{x+m}$} obtained in $d_i \in D$ do
    Let $as_i$ be the average review sentiment for $d_i$;
    $as_i \leftarrow$ len([$s$ in {$s_x$, $s_{x+1}$, ... $s_{x+m}$} if $s$ is negative])/m
end
for each $d_i \in D$ do
    Let $sc_i$ be the sentiment change between $d_i$ and $d_{i+1}$;
    $sc_i \leftarrow (as_{i+1} - as_i)$
end
Let $X$ be the continuous random variable of sentiment changes;
Then, as $X \sim EPD(\mu, \alpha, \beta)$, calculate $\mu, \alpha,$ and, $\beta$;
Let $AD$ be the set of abnormal sentiment change time spans;
$AD \leftarrow$ [for $sc_i$ in $X$ if $sc_i > \mu + 3\sigma$];
if $AD$ exist then
    Let $U$ be the set of updates released within the fixed period;
    $Ut \leftarrow$ [the update content of $u$ for $u$ in $U$];
    for each $ad_i \in AD$ do
        $F \leftarrow$ sorted(getTF-IDFList([reviews in $d_{i+1}$]), reverse=True)[:150];
        $K \leftarrow$ [getKeywords($ut$) for $ut$ in $Ut$];
        if the $u_i$ with the max([Similarity(k, F) for k in K]) is the nearest previous update then
            return $u_i$;
        else
            $ad_i$ is not caused by update;
        end
    end
else
    no abnormal days detected
end
```

Let R be a set of user reviews for a particular mobile app A within a defined time period, where each individual review $r_i \in R$ is tagged with a specific time point. Meanwhile, let U also be the set of updates released by the developers of app A within the same time period, where each update $u_i \in U$ is also released at a particular time point. Thus, if any abnormal days are detected, for each identified day with abnormal sentiment change during the period, ad_i, we can verify whether it is the nearest previous update that is problematic and causing ad_i. Herein, the similarity between the negative reviews and the previous update is calculated by the average of the Word2Vec similarity values of each update description keyword and each of the top TF-IDF review keywords. The according TF-IDF are also taken into account as the weight of the similarity values. To be noted, we hereby select 150 keywords with the highest TF-IDF score to ease the calculation cost of the algorithm, while maintaining its accuracy. If based on the comparison of similarities, the nearest previous update is not identified as problematic, the negativity in user reviews can be caused by other issues, which requires further investigation.

4. Case Study

This case study is to validate the usefulness of the proposed method in identifying the abnormal days from user reviews and the problematic updates during the mobile app maintenance. We collect the review data from five popular mobile apps and apply our method. The result shows that the problematic updates can be identified when using this method.

4.1. Data Preprocessing

Preprocessing on the acquired raw review data is required before experimenting with the proposed method. We hereby apply the following steps to clean the data into usable.

Filtering non-English reviews. We screen out the non-English review sentences using Langdetect [38], a convenient language detecting package for Python language. Langdetect identifies the language of a particular sentence using the sentence as a whole instead of individual words within. Hence, due to the nature of user reviews, the sentences with misspelled words, slurs and abbreviations, used often in social media, will not be filtered out.

Separating Long Reviews into Sentences. Despite mobile app reviews being shorter, in general, compared with reviews on other platforms, particular reviews still contain multiple sentences, each of which might convey different meanings and sentiments. Therefore, we use the sentence tokenizer feature from the NLTK [39] python package to obtain the sentence set of each individual review item.

Calculating the Sentiments. Herein, we use the Valence Aware Dictionary for sEntiment Reasoning (VADER) approach [40] to calculate the sentiment score of each review sentence. VADER is a commonly used sentiment analysis tool due to its classification accuracy on sentiment towards positive, negative and neutral classes, which is even higher than individual human raters in the social media domain. According to Hutto and Gilbert's experiment results [40], the F1 score of VADER on social media text (i.e., short text with informal language) is 0.96. Due to the unique trait of mobile app reviews being short and informal compared to other reviews types (e.g., movie reviews), we expect such sentiment analysis being as accurate as that on social media text. To further verify the accuracy of the VADER approach on mobile app reviews, we calculate the precision, recall and F1 score. Shown in Table 1, the overall accuracy of VADER on mobile app review sentences is 0.819. When considering multi-sentences reviews, we calculate the sentiment score of each review as the mean of scores of each sentence in the review. Shown in Table 1, the accuracy of overall accuracy of VADER on mobile app reviews is 0.842.

Table 1. Sentiment score accuracy testing.

App	Review-Level			Sentence-Level		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Imo	0.826	0.786	0.804	0.822	0.777	0.797
Hangouts	0.839	0.803	0.819	0.802	0.780	0.790
Messenger	0.883	0.835	0.855	0.869	0.815	0.836
Skype	0.874	0.828	0.844	0.828	0.790	0.795
Whatsapp	0.851	0.814	0.830	0.833	0.787	0.805
Overall	0.868	0.823	0.842	0.850	0.800	0.819

In addition, compared to the other popular sentiment analysis approaches, such as SenticNet [41], SentiWordNet [42], Affective Norms for English Words [43] and Word-Sense Disambiguation [44], its overall classification accuracy on product reviews from Amazon, movie reviews and editorials from NYTimes also prevails. Furthermore, VADER is easy to import and use as it is integrated in the NLTK package.

Filtering Stopwords and Lemmatization. Every review sentence shall contain words that are used often but carry less meaning, i.e., the stopwords. The stopwords must be removed in order to obtain meaningful term frequency results. The adopted stopwords set also includes the app-specific terms with high occurence, like *'app'*, *'application'* and *'whatsapp'*, as well as the common typos of other stopwords, like *'dont'*, *'whasapp'* and *'appp'*. In addition, the acquired review sentences are also transformed into lower cases and lemmatized before the calculation of term frequency so that the same term with different cases shall be seen as one. We hereby use the stopword set in NLTK corpus package to screen the stopwords from review sentences, and the *WordNetLemmatizer* from NLTK stem package to lemmatize words.

Selecting Nouns and Verbs. Nouns and verbs are two major word types we consider in term frequency analysis. Adjectives and adverbs are filtered because the term frequency analysis is to gain an insight into the issues in reviews with a negative sentiment score. The adjectives and adverbs which mainly contribute to the sentiment score have been considered in sentiment analysis, and require no redundant covering in the term frequency analysis. We use the RegexpTokenizer from NLTK and the pos_tags of the tokenized words to identify and filter words that are not nouns or verbs.

4.2. Data Description

In this study, we use the user reviews of five instant messenger mobile apps from Android platform, namely Imo, Hangouts, Messenger, Skype and Whatsapp. We collect for each app the reviews from 1 September 2016 to 31 August 2017, eliminate the non-English reviews and tokenize each into sentences. The numbers of reviews and the obtained English reviews and numbers of sentences for each app are shown in Table 2, while the number of review sentences on each day is shown in Figure 3-left. Together with the user reviews, we also collect the release date information regarding the updates within the given period. The numbers of updates counted for each app are also shown in Table 2.

Table 2. Application (app) review statistics.

App Name	Reviews	English Reviews	Sentences	Updates
Imo	202,870	86,194	100,838	84
Hangouts	122,622	68,535	10,1704	43
Messenger	1,654,360	886,643	1,185,368	105
Skype	153,128	105,875	189,995	76
Whatsapp	1,660,145	851,662	109,8583	49
total	3,793,125	1,998,909	2,676,488	357

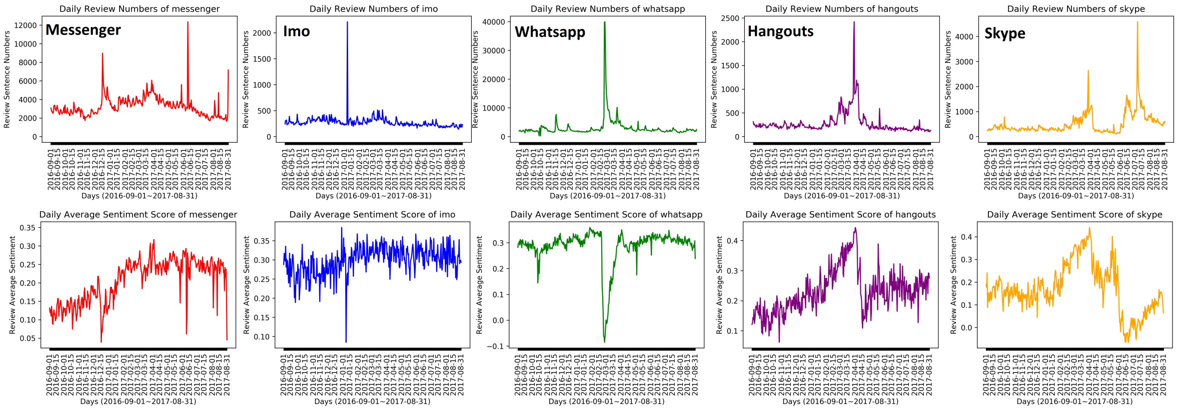


Figure 3. Number of reviews and average sentiment score by day (from 2016-09-01 to 2017-09-01).

Thereafter, for each of the 2,676,488 review sentences, we calculate its sentiment score using the VADER sentiment algorithm. The sentiment scores range from 1.0 (i.e., very positive) to −1.0 (i.e., very negative). Furthermore, we calculate the daily average sentiment score for each selected app and plot the changes in Figure 3. Such average sentiment score is calculated as follows. Provided m reviews are given on the day d_i with the sentiment score of each review r_i within is calculated and denoted as

s_i ($i = 1, 2, \ldots, m$), the average sentiment score on d_i is calculated as $(\sum_{i=1}^{m} s_i)/m$. From the sentiment changes, we can observe that the majority of the daily sentiment range from 0.1 to 0.4. It indicates that for these five apps, the overall sentiment within the given period is still positive despite such changes. In addition, we find that the daily average rating of each app is highly correlated with the daily average sentiment with an average Pearson R score of 0.93. It indicates the sentiment of the reviews can be used to reflect the users' evaluation to their general fondness of the apps.

4.3. Results

Herein, we apply the previously presented method, that is, identifying abnormal days based on the distribution of negative sentiment changes and matching such abnormality to a particular update, on each review dataset of the given five mobile applications. By doing so, we aim to investigate whether such a method can be used towards obtaining meaningful results.

4.3.1. Identify Abnormal Days

For the 1-year review sentiment data of the given five mobile apps, we firstly calculated the sentiment changes of each day as the negative review proportion change. Considering all 1840 obtained daily sentiment change values as the continuous random variable, we detect the most likely distribution model that fits the values. By doing so, we find that the best fitting distribution model is the exponential power distribution (EPD), with a p value of 0.968. Furthermore, the parameters for the obtained EPD model are $\mu = 0.0002$, $\alpha = 0.0227$ and $\beta = 1.0444$.

Based on the obtained distribution parameters, we calculate the confidential intervals ($\mu \pm 3\sigma$) for each set of sentiment change values for each mobile app (shown in Table 3). Based on the obtained confidential intervals, we can identify the abnormal days of each mobile app, where the sentiment changes are greater than $\mu + 3\sigma$.

Table 3. Confidential intervals and identified abnormal days.

App Name	CI	Abnormal Days (Year-Month-Day)
Imo	(−0.082, 0.083)	['13 December 2016', '7 January 2017', '3 August 2017']
Hangouts	(−0.098, 0.099)	[ ]
Messenger	(−0.060, 0.061)	['16 December 2016', '9 June 2017', '3 August 2017', '31 August 2017']
Skype	(−0.095, 0.096)	['4 September 2016', '21 May 2017', '25 May 2017']
Whatsapp	(−0.053, 0.052)	['12 October 2016', '15 October 2016', '21 February 2017', '23 February 2017', '24 February 2017', '3 May 2017']

By observing the daily sentiment changes, we can easily map the obtained abnormal days results with the significant sentiment changes from the evolution chart. Figure 4-left shows the review sentiment changes of Whatsapp. The x-axis represents the consecutive dates, while the y-axis represents the proportion of positive, neutral and negative reviews (shown in green, blue and red curves, respectively). The six abnormal days can be observed by the obvious rise in negative sentiment and fall of positive sentiment. Furthermore, such abnormal days can also be validated by the changes of top frequent words of each day. By comparing the Jaccard similarity of the top frequent words of each day, we can also find that the top frequent words of abnormal days are largely different from those of other days. The similarity values of the top 10 frequent words between each day for Whatsapp are shown in Figure 4-right, where the dark color indicates the low similarity value. Therein, we can observe that the identified abnormal days contain different top frequent words from the other days, when the reviews of the other days largely share common top frequent words.

In addition, we can also observe that all the days on which top frequent words are different from those of the others are not identified as abnormal days. For example, from 2017-02-20 to 2017-03-31, the daily average review sentiments of Whatsapp are continuously more negative than usual, despite the rising from 2017-02-27. Accordingly, we can observe that the similarities among the top frequent words during that period are high, but also low towards other days. Thus, an implication can be made that the negative effect of the identified abnormal days lasts for the whole period regarding similar issues.

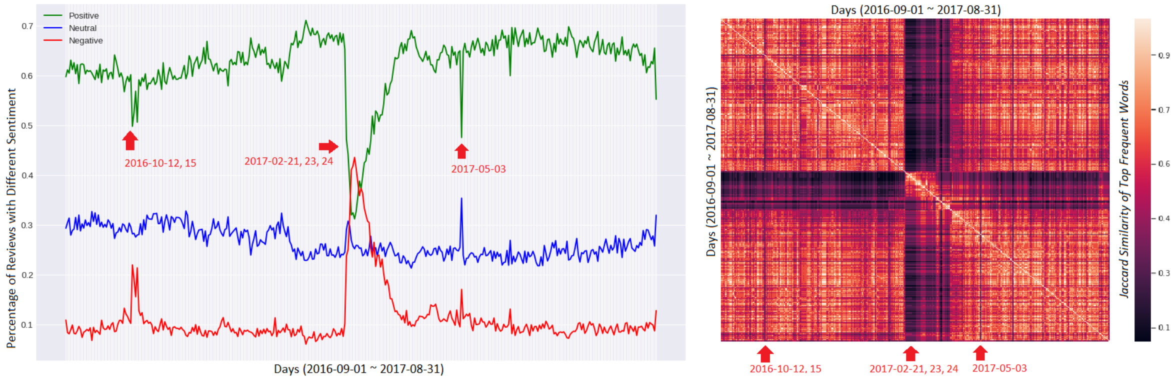
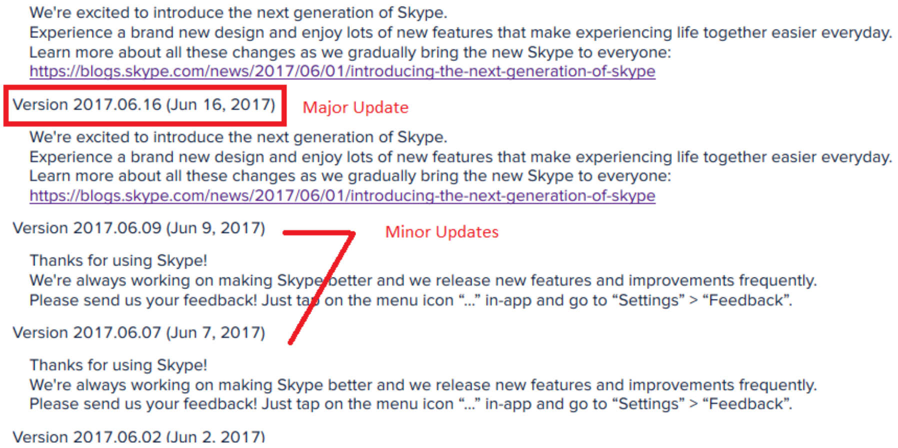


Figure 4. The identified abnormal days of Whatsapp as an example.

4.3.2. Identify Problematic Updates

Based on the obtained abnormal days of the given mobile apps, we continue to investigate the particular updates that result in such abnormality in review sentiment, if any. For such purpose, we retrieve the update description text of each update during the review period for the mobile apps. Herein, the update description texts are required to specifically describe, to a certain extent, the update content for each particular new version. Therefore, we only select Whatsapp and Skype for this experiment, due to the fact that the update description texts of IMO and Messenger are vague and identical throughout the period, and no abnormal days are identified for Hangouts. Furthermore, we take into account only the major updates, which are identified as the first updates whose description text is different from that of the previous one(s). For example, shown in Figure 5, Version 2017.06.16 is identified as a major update when the other ones are identified as minor ones.


We're excited to introduce the next generation of Skype.
Experience a brand new design and enjoy lots of new features that make experiencing life together easier everyday.
Learn more about all these changes as we gradually bring the new Skype to everyone:
https://blogs.skype.com/news/2017/06/01/introducing-the-next-generation-of-skype

Version 2017.06.16 (Jun 16, 2017) Major Update

We're excited to introduce the next generation of Skype.
Experience a brand new design and enjoy lots of new features that make experiencing life together easier everyday.
Learn more about all these changes as we gradually bring the new Skype to everyone:
https://blogs.skype.com/news/2017/06/01/introducing-the-next-generation-of-skype

Version 2017.06.09 (Jun 9, 2017) Minor Updates

Thanks for using Skype!
We're always working on making Skype better and we release new features and improvements frequently.
Please send us your feedback! Just tap on the menu icon "..." in-app and go to "Settings" > "Feedback".

Version 2017.06.07 (Jun 7, 2017)

Thanks for using Skype!
We're always working on making Skype better and we release new features and improvements frequently.
Please send us your feedback! Just tap on the menu icon "..." in-app and go to "Settings" > "Feedback".

Version 2017.06.02 (Jun 2, 2017)

Figure 5. Examples of major and minor updates.

In order to identify the cause of each identified abnormal day, we compare the similarity between the negative review content of the identified abnormal days and the description text of the major updates. As mentioned in the prior section, we select only nouns and verbs from both the update description and reviews. Furthermore, for the review texts, we select only the words with high TF-IDF scores, which represent the main content of the text.

Figure 6 shows the similarity of the negative reviews of each identified abnormal day and the descriptions of the nine major updates. Therein, the nearest previous update of each abnormal day is marked red. As shown in the figure, for the identified abnormal days of Whatsapp: 2017-02-21, 2017-02-23, 2017-02-24 and 2017-05-03, the similarities of the negative reviews and the descriptions of their nearest previous updates are significantly high. By observing the review texts of these four dates, we locate 20017 negative review sentences out of 32,922 containing the keyword "update" or "version", which shows the connection between the review negativity of the abnormal days and the most recent updates. However, for the abnormal days 2016-10-12 and 2016-10-15, the similarities are not as significant as with the four ones shown in the first two charts of Figure 6. We investigate closely on the update description text of Version 2016.10.11, stating "*Now you can draw or add text and emojis to photos and videos you capture within WhatsApp...New emoji. And sending a single emoji will now appear larger in chats*". As we find "emoji" being the core feature updated in this particular version, 46 out of 105 negative review sentences on these two days contain the word "update", "version" or "emoji".

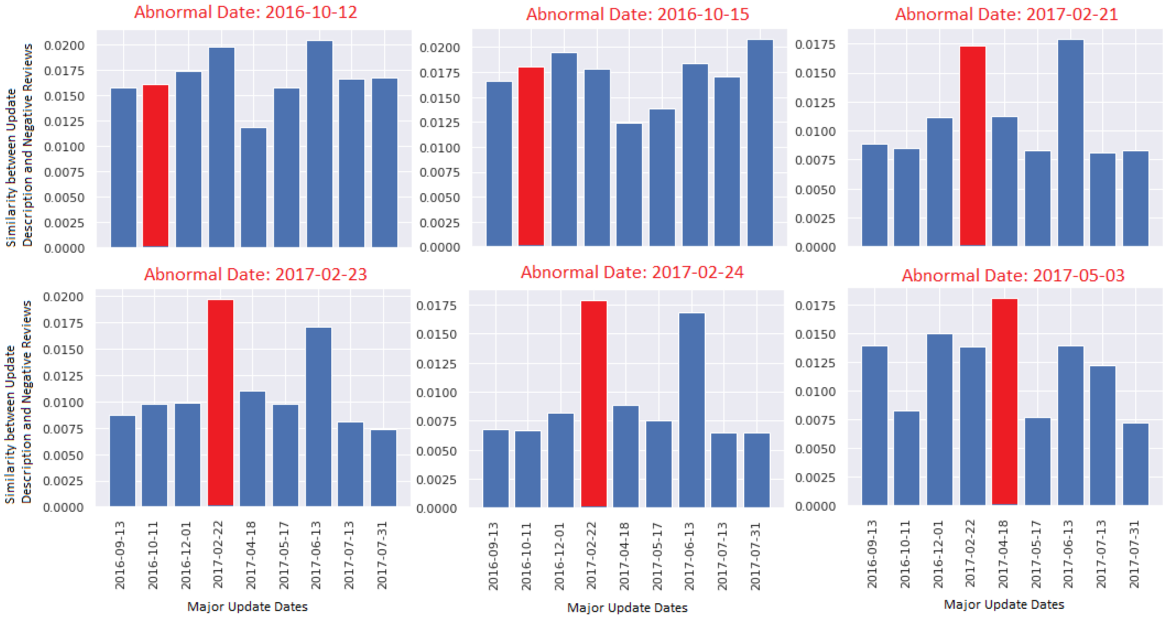


Figure 6. Similarity between update description and negative reviews of abnormal days for Whatsapp.

The result for Skype is shown in Figure 7, which is not significant compared to that for Whatsapp. The negative reviews of the identified abnormal days, 2017-05-21 and 2017-05-25, cannot be matched to their nearest previous update by similarities to the description text. Abnormal day 2016-09-04 is ignored here, as it occurs before the first update retrieved within the review data period. By further investigating the review texts of the two abnormal days, we find only 4 out of 70 review sentences contain word "update" or "version". The review texts are mostly describing quality related issues with scattered topics, which indicates that it is not the particular update that cause the negativity in reviews.

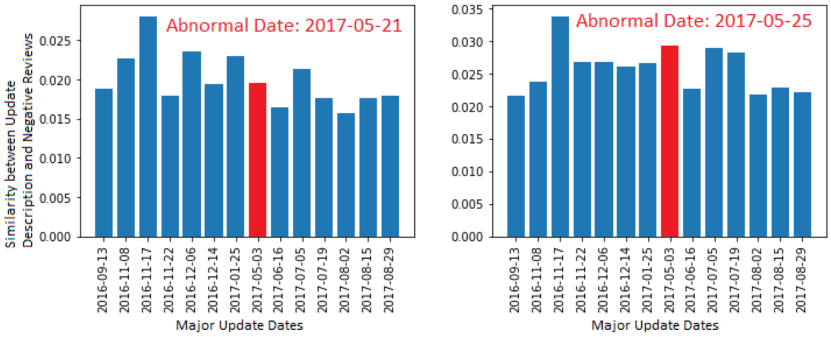


Figure 7. Similarity between update description and negative reviews of abnormal days for Skype.

To summarize, for a particular mobile app, the proposed method can be used to identify specific time period, e.g., days, during which the negative sentiment of the reviews is considered abnormal compared to a much larger set of reviews within a much longer period. It is also possible to identify the particular update, mostly the nearest one before an identified abnormal day, that causes such abnormality in review sentiment (e.g., the problematic Version 2017.02.22 of Whatsapp). However, when not the nearest update is identified with the highest similarity between negative reviews and the update description texts, it is likely that the negative reviews of the abnormal days are regarding more general and scattered issues, instead of a particular update (e.g., the result of Skype). The vague description of the update content and the limited review numbers can be the factors that result in various unpredictable outcomes with the method.

5. Discussion

Compared to the previous studies on mobile app review analysis [21], this study focuses on finding the negative user reviews in a particular day that matter the most to the developers in terms of sentiment changes, and investigating the connection between these reviews and the potential updates that cause the negativity. It is a light-weight method that eases the effort of the continuous analysis of the increasing amount of user reviews that also contain large volume of non-informative content. Regarding the mobile app maintenance practice, the method can facilitate the emergency-oriented maintenance model, which is usually adopted by the mobile apps with stable core functionalities [45]. Thus, our method can be considered more suitable for indie app developers [46] or a small development team to swiftly identify and fix problematic updates reflected by noticeable amount of negativity in reviews.

Taking into consideration the results of the case study above, the method performs well when the volume of review data is sufficient and the update description texts are detailed composed. Hence, one of the limitations of this study is the validation of the method towards particular mobile app cases, which receive only a limited amount of reviews. Furthermore, due to the adoption of distribution analysis, the method can also suffer from a “cold start” issue, that is, for a newly launched app product without sufficient review data, the method will fail to perform. Another critical limitation of this research is the threat to validity, as only the review data of five instant messenger mobile apps are taken into account. Thus, further verification of the method regarding its prerequisite on review data volume and update description texts is required. In addition, a further validation of the review sentiment change distribution model is also necessary, which can be done with a larger mobile app data repository. Accuracy of the sentiment analysis can, to a certain extent, influence the validity of the identified abnormal days. Thus, the adoption of such method on the analysis of other reviews, e.g., movie reviews, product reviews, etc., shall yield to such threat to validity, especially when the accuracy is relatively low and the volume of the data is limited.

In our future work, we will focus on addressing the limitations mentioned above and improving the method on the following aspects. Methods, such as aspect-based sentiment analysis [47] together with user review feature extraction [24], will largely enhance the effectiveness of the method in terms of the uneven user sentiment on various complaint types from mobile app users [2]. Similarly, topic model techniques can also be applied to extract app feature related topics, which will also enhance the usefulness of the method. Building on such results, the method will be able to prioritize the severity of topic-based or feature-based categorized issues and facilitate developers planning on updates. In addition, taking into account the different reviewing behaviors of app users, as well as their preferences on app types, can also provide insights on analyzing the helpfulness of the reviews given. On the other hand, a method for automatically evaluating the helpfulness of update description and extracting the features of the updates shall also be helpful towards reducing the human effort provided the number of apps selected for the future work being excessive.

6. Conclusions

This study proposes a sentiment-statistical approach for detecting abnormal days during mobile app maintenance. The core of the method is based on the analysis of review sentiment distribution, and the similarities between update descriptions and review texts. Specifically, critical conclusions can be made when the negative sentiment increases sharply in a particular time period. In addition, we use the same method to map abnormal days to potential updates that cause such days.

Our method aims to facilitate mobile app developers to identify the critical moment during mobile app evolution when the users' opinion towards the app product grows overwhelmingly negative very quickly, and checking whether such negativity is caused by the nearest update. The results of the case study show that the proposed method performs effectively, however, with the prerequisite of having sufficient volume of user review data and adequately detailed update description text.

Author Contributions: Conceptualization, X.L., B.Z., Z.Z. and K.S.; data curation, X.L.; formal analysis, X.L. and B.Z.; investigation, X.L., Z.Z. and K.S.; methodology, X.L.; software, X.L.; supervision, Z.Z. and K.S.; validation, X.L.; visualization, X.L.; writing—original draft, X.L. and B.Z.; writing—review and editing, X.L., Z.Z. and K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nayebi, M.; Adams, B.; Ruhe, G. Release Practices for Mobile Apps–What do Users and Developers Think? In Proceedings of the 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (Saner), Osaka, Japan, 14–18 March 2016; Volume 1, pp. 552–562.
2. Khalid, H.; Shihab, E.; Nagappan, M.; Hassan, A.E. What do mobile app users complain about? *IEEE Softw.* **2014**, *32*, 70–77.
3. Holzer, A.; Ondrus, J. Mobile application market: A developer's perspective. *Telemat. Inf.* **2011**, *28*, 22–31.
4. Galvis Carreño, L.V.; Winbladh, K. Analysis of user comments: an approach for software requirements evolution. In Proceedings of the 2013 International Conference on Software Engineering, San Francisco, CA, USA, 18–26 May 2013; pp. 582–591.
5. Chen, N.; Lin, J.; Hoi, S.C.; Xiao, X.; Zhang, B. AR-miner: mining informative reviews for developers from mobile app marketplace. In Proceedings of the 36th International Conference on Software Engineering, Hyderabad, India, 31 May 31–7 June 2014; pp. 767–778.
6. Guzman, E.; Maalej, W. How do users like this feature? a fine grained sentiment analysis of app reviews. In Proceedings of the 2014 IEEE 22nd international requirements engineering conference (RE), Karlskrona, Sweden, 25–29 August 2014; pp. 153–162.
7. Maalej, W.; Nabil, H. Bug report, feature request, or simply praise? on automatically classifying app reviews. In Proceedings of the 2015 IEEE 23rd international requirements engineering conference (RE), Ottawa, ON, Canada, 24–28 August 2015; pp. 116–125.
8. Panichella, S.; Di Sorbo, A.; Guzman, E.; Visaggio, C.A.; Canfora, G.; Gall, H.C. How can i improve my app? classifying user reviews for software maintenance and evolution. In Proceedings of the 2015 IEEE international conference on software maintenance and evolution (ICSME), Bremen, Germany, 27 September–3 October 2015; pp. 281–290.
9. McIlroy, S.; Ali, N.; Khalid, H.; Hassan, A.E. Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. *Empir. Softw. Eng.* **2016**, *21*, 1067–1106.
10. Li, X.; Zhang, Z.; Stefanidis, K. Sentiment-aware Analysis of Mobile Apps User Reviews Regarding Particular Updates. In Proceedings of the Thirteenth International Conference on Software Engineering Advances (ICSEA), Nice, France, 14–18 October 2018; p. 109.
11. Fu, B.; Lin, J.; Li, L.; Faloutsos, C.; Hong, J.; Sadeh, N. Why people hate your app: Making sense of user feedback in a mobile app store. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–13 August 2013; pp. 1276–1284.
12. Zhang, L.; Hua, K.; Wang, H.; Qian, G.; Zhang, L. Sentiment analysis on reviews of mobile users. *Procedia Comput. Sci.* **2014**, *34*, 458–465.

13. Fafalios, P.; Iosifidis, V.; Stefanidis, K.; Ntoutsi, E. Multi-aspect Entity-Centric Analysis of Big Social Media Archives. In Proceedings of the Research and Advanced Technology for Digital Libraries—21st International Conference on Theory and Practice of Digital Libraries TPDL, Thessaloniki, Greece, 18–21 September 2017; pp. 261–273.
14. Stratigi, M.; Li, X.; Stefanidis, K.; Zhang, Z. Ratings vs. Reviews in Recommender Systems: A Case Study on the Amazon Movies Dataset. In *European Conference on Advances in Databases and Information Systems*; Springer: Cham, Switzerland, 2019.
15. Villarroel, L.; Bavota, G.; Russo, B.; Oliveto, R.; Di Penta, M. Release planning of mobile apps based on user reviews. In Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), Austin, TX, USA, 14–22 May 2016; pp. 14–24.
16. Ciurumelea, A.; Schaufelbühl, A.; Panichella, S.; Gall, H.C. Analyzing reviews and code of mobile apps for better release planning. In Proceedings of the 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), Klagenfurt, Austria, 20–24 February 2017; pp. 91–102.
17. Palomba, F.; Salza, P.; Ciurumelea, A.; Panichella, S.; Gall, H.; Ferrucci, F.; De Lucia, A. Recommending and localizing change requests for mobile apps based on user reviews. In Proceedings of the 39th International Conference on Software Engineering, Buenos Aires, Argentina, 20–28 May 2017; pp. 106–117.
18. Koskela, M.; Simola, I.; Stefanidis, K. Open Source Software Recommendations Using Github. In Proceedings of the Digital Libraries for Open Knowledge, 22nd International Conference on Theory and Practice of Digital Libraries TPDL, Porto, Portugal, 10–13 September 2018; pp. 279–285.
19. Möller, A.; Michahelles, F.; Diewald, S.; Roalter, L.; Kranz, M. Update behavior in app markets and security implications: A case study in google play. In Proceedings of the 14th International Conference on Human Computer Interaction with Mobile Devices and Services, San Francisco, CA, USA, 2012; pp. 3–6.
20. Xia, X.; Shihab, E.; Kamei, Y.; Lo, D.; Wang, X. Predicting crashing releases of mobile applications. In Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Ciudad Real, Spain, 8–9 September 2016; p. 29.
21. Genc-Nayebi, N.; Abran, A. A systematic literature review: Opinion mining studies from mobile app store user reviews. *J. Syst. Softw.* **2017**, *125*, 207–219.
22. Gao, C.; Xu, H.; Hu, J.; Zhou, Y. Ar-tracker: Track the dynamics of mobile apps via user review mining. In Proceedings of the 2015 IEEE Symposium on Service-Oriented System Engineering, San Francisco, CA, USA, 30 March 30–3 April 2015; pp. 284–290.
23. Chandy, R.; Gu, H. Identifying spam in the iOS app store. In Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality, Lyon, France, 16 April 2012; pp. 56–59.
24. Vu, P.M.; Nguyen, T.T.; Pham, H.V.; Nguyen, T.T. Mining user opinions in mobile app reviews: A keyword-based approach (t). In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), Lincoln, NA, USA, 9–13 November 2015; pp. 749–759.
25. Iacob, C.; Harrison, R. Retrieving and analyzing mobile apps feature requests from online reviews. In Proceedings of the 10th Working Conference on Mining Software Repositories, San Francisco, CA, USA, 18–19 May 2013; pp. 41–44.
26. Pagano, D.; Maalej, W. User feedback in the appstore: An empirical study. In Proceedings of the 2013 21st IEEE international requirements engineering conference (RE), Rio de Janeiro, Brazil, 15–19 July 2013; pp. 125–134.
27. Zimina, E.; Nummenmaa, J.; Järvelin, K.; Peltonen, J.; Stefanidis, K.; Hyyrö, H. GQA: Grammatical Question Answering for RDF Data. In Proceedings of the Semantic Web Challenges—5th SemWebEval Challenge at ESWC, Heraklion, Greece, 3–7 June 2018; pp. 82–97.
28. Wang, S.; Wang, Z.; Xu, X.; Sheng, Q.Z. App Update Patterns: How Developers Act on User Reviews in Mobile App Stores. In *International Conference on Service-Oriented Computing*; Springer: Berlin, Germany, 2017; pp. 125–141.
29. Li, X.; Zhang, Z.; Stefanidis, K. Mobile App Evolution Analysis Based on User Reviews. In Proceedings of the International Conference on Intelligent Software Methodologies, Tools, and Techniques, Granada, Spain, 26–28 September 2018; pp. 773–786.
30. Massey, F.J., Jr. The Kolmogorov-Smirnov test for goodness of fit. *J. Am. Stat. Assoc.* **1951**, *46*, 68–78.
31. Nadarajah, S. A generalized normal distribution. *J. Appl. Stat.* **2005**, *32*, 685–694.
32. Pukelsheim, F. The three sigma rule. *Am. Stat.* **1994**, *48*, 88–91.

33. Chebyshev, P.L. Des valeurs moyennes, Liouville's. *J. Math. Pures Appl.* **1867**, *12*, 177–184.
34. Hoon, L.; Vasa, R.; Schneider, J.G.; Grundy, J. *An Analysis of the Mobile App Review Landscape: Trends And Implications*; Faculty of Information and Communication Technologies, Swinburne University of Technology: Melbourne, Australia, 2013.
35. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2013; pp. 3111–3119.
36. Xue, B.; Fu, C.; Shaobin, Z. A study on sentiment computing and classification of sina weibo with word2vec. In Proceedings of the 2014 IEEE International Congress on Big Data, Anchorage, AK, USA, 27 June–2 July 2014; pp. 358–363.
37. Ramos, J. Using tf-idf to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning, Piscataway, NJ, USA, 3–8 December 2003; Volume 242, pp. 133–142.
38. Langdetect. Available online: https://pypi.python.org/pypi/langdetect (accessed on 30 September 2019).
39. NLTK. Available online: http://www.nltk.org (accessed on 30 September 2019).
40. Hutto, C.J.; Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the Eighth international AAAI conference on Weblogs and Social Media, Ann Arbor, MI, USA, 1–4 June 2014.
41. Cambria, E.; Speer, R.; Havasi, C.; Hussain, A. Senticnet: A publicly available semantic resource for opinion mining. In Proceedings of the AAAI Fall Symposium: Commonsense Knowledge, Arlington, VA, USA, 2010; Volume 10.
42. Baccianella, S.; Esuli, A.; Sebastiani, F. *Sentiwordnet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining*; Lrec: Baton Rouge, Louisiana, 2010; Volume 10, pp. 2200–2204.
43. Bradley, M.M.; Lang, P.J. *Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings*; Technical Report; University of Florida, The Center for Research in Psychophysiology: Gainesville, FL, USA, 1999.
44. Stevenson, M.; Wilks, Y. Word sense disambiguation. In *The Oxford Handbook of Computational Linguistics*; Oxford University Press: Oxford, UK, 2003; pp. 249–265.
45. Li, X.; Zhang, Z.; Nummenmaa, J. Models for mobile application maintenance based on update history. In Proceedings of the 2014 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), Lisbon, Portugal, 28–30 April 2014; pp. 1–6.
46. Qiu, Y.; Gopal, A.; Hann, I.H. Logic pluralism in mobile platform ecosystems: A study of indie app developers on the iOS app store. *Inf. Syst. Res.* **2017**, *28*, 225–249.
47. Thet, T.T.; Na, J.C.; Khoo, C.S. Aspect-based sentiment analysis of movie reviews on discussion boards. *J. Inf. Sci.* **2010**, *36*, 823–848.