INFAR: Insight Extraction from App Reviews

Cuiyun Gao
Shenzhen Research Institute
The Chinese University of Hong Kong
Shenzhen, China
cygao@cse.cuhk.edu.hk

Jichuan Zeng
Shenzhen Research Institute
The Chinese University of Hong Kong
Shenzhen, China
jczeng@cse.cuhk.edu.hk

David Lo
Singapore Management University
Singapore
davidlo@smu.edu.sg

Chin-Yew Lin
Microsoft Research
Beijing, China
cyl@microsoft.com

Michael R. Lyu
The Chinese University of Hong Kong
Hong Kong, China
lyu@cse.cuhk.edu.hk

Irwin King
The Chinese University of Hong Kong
Hong Kong, China
king@cse.cuhk.edu.hk

ABSTRACT

App reviews play an essential role for users to convey their feedback about using the app. The critical information contained in app reviews can assist app developers for maintaining and updating mobile apps. However, the noisy nature and large-quantity of daily generated app reviews make it difficult to understand essential information carried in app reviews. Several prior studies have proposed methods that can automatically classify or cluster user reviews into a few app topics (*e.g.*, security). These methods usually act on a static collection of user reviews. However, due to the dynamic nature of user feedback (*i.e.*, reviews keep coming as new users register or new app versions being released) and multiple analysis dimensions (*e.g.*, review quantity and user rating), developers still need to spend substantial effort in extracting contrastive information that can only be teased out by comparing data from multiple time periods or analysis dimensions. This is needed to answer questions such as: what kind of issues users are experiencing most? is there an unexpected rise in a particular kind of issue? etc. To address this need, in this paper, we introduce **INFAR**, a tool that automatically extracts **IN**sights **F**rom **A**pp **R**eviews across time periods and analysis dimensions, and presents them in natural language supported by an interactive chart. The insights INFAR extracts include several perspectives: (1) salient topics (*i.e.*, issue topics with significantly lower ratings), (2) abnormal topics (*i.e.*, issue topics that experience a rapid rise in volume during a time period), (3) correlations between two topics, and (4) causal factors to rating or review quantity changes. To evaluate our tool, we conduct an empirical evaluation by involving six popular apps and 12 industrial practitioners, and 92% (11/12) of them approve the practical usefulness of the insights summarized by INFAR.
Demo Tool Website:https://remine-lab.github.io/paper/infar.html
Demo Video: https://youtu.be/MjcoiyjA5TE

ESEC/FSE '18, November 4–9, 2018, Lake Buena Vista, FL, USA

ACM ISBN 978-1-4503-5573-5/18/11…$15.00
https://doi.org/10.1145/3236024.3264595

CCS CONCEPTS

• **Software and its engineering** → *Software functional properties*;
• **Information systems** → *Information integration*;

KEYWORDS

App review, review topic, insight extraction

ACM Reference Format:
Cuiyun Gao, Jichuan Zeng, David Lo, Chin-Yew Lin, Michael R. Lyu, and Irwin King. 2018. INFAR: Insight Extraction from App Reviews. In *Proceedings of the 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '18), November 4–9, 2018, Lake Buena Vista, FL, USA.* ACM, New York, NY, USA, 4 pages.
https://doi.org/10.1145/3236024.3264595

1 INTRODUCTION

User reviews play an important role in mobile software development. They serve as an essential channel between app developers and users, and deliver users' recent experience with the apps. By analyzing app reviews, developers can gain valuable information for app updates, including the features to improve, new functionalities sought-after by users, and also functional and non-functional issues to be rectified. Thus, app review analysis is one step that can be highly beneficial in agile mobile app development.

Despite its benefits, analyzing app reviews often poses a challenge to developers. The number of such reviews is often voluminous – which is true especially for popular apps. To deal with this challenge, several studies have proposed methods to help developers better manage app reviews. These include studies that categorize app reviews into several topics [3], prioritize reviews of different topics [5, 2, 6], or allow developers to search for reviews of interest given some keywords [9]. Although the current techniques assist in review analysis, they are mostly focusing on analyzing a static review collection and provide little support for contrasting reviews across multiple time periods and dimensions. Developers need to put non-trivial amount of effort to produce contrastive insights from results of existing tools.

As an example, consider Developer A who is responsible for analyzing user reviews and reporting critical user feedback to other developers. Let's consider her employing SURF [3], a popular review

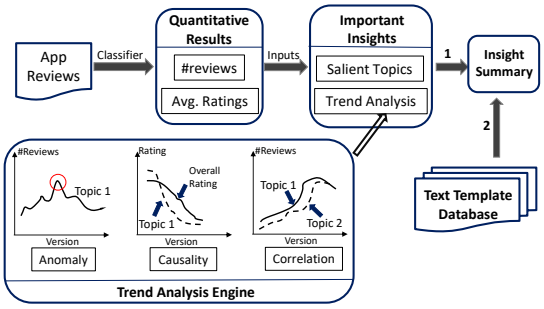


Figure 1: Workflow of INFAR.

classification tool, to classify the collected reviews into topics[1]. With the categorization of reviews, she is able to identify reviews for a given topic. However, a non-trivial amount of manual work is still needed to answer questions such as: what are the abnormal topics in the current version? which topic impacts a significant decrease in current review rating? etc. These questions are important for developers to take corrective actions to improve the satisfaction of app users, and they cannot be derived easily using SURF, which does not support analysis comparing reviews from multiple time periods.

In the paper, we aim to fill the gap of existing research by distilling insights that are not produced by existing tools. Specifically, we build a tool to extract four types of contrastive insights from different time periods and two analysis dimensions (*i.e.*, review quantity and user rating). The insights include: (1) salient topics (*i.e.*, issue topics with significantly lower ratings), (2) abnormal topics (*i.e.*,issue topics that experience a rapid rise in volume during a time period), (3) correlations between two topics, and (4) causal factors to rating or review quantity changes. We name our tool **INFAR**, for automatic **IN**sight extraction **F**rom **A**pp **R**eviews. To evaluate our tool, we conduct an empirical evaluation by involving six popular apps and 12 industrial practitioners, and 92% (11/12) of them approve the practical usefulness of the insights summarized by INFAR.

2 METHODOLOGY

Figure 1 depicts the workflow of INFAR, which consists of two major steps. With raw reviews as input, INFAR first preprocesses the raw reviews and classifies the reviews according to predefined topics. Given the classification results, INFAR captures four types of insights and explains the insights with chart visualization. The insight types are salient topics ($\mathcal{S}$), abnormal topics ($\mathcal{A}$), causal factors ($\mathcal{Y}$), and correlated topics ($\mathcal{E}$), among which the latter three are extracted by a trend analysis engine. After these insights are extracted, INFAR presents them in natural language using a set of text templates. In this paper, we employ SURF [3] as the classifier to group reviews into intention types and topic clusters. For simplicity, we refer to these as topics in this paper.

[1]SURP defines 12 review topics and four types. The 12 topics are App, GUI, Contents, Pricing, Feature or Functionality, Improvement, Updates/Versions, Resources, Security, Download, Model, and Company. The types include Bug, Request, Question, and Info.

2.1 Insight Extraction

We define insight scores to measure each insight's significance to developers, denoted as $\text{Sig}_{\mathcal{T}}$ (where $\mathcal{T}$ is the insight type). We use $X = \{x_1, x_2, ..., x_m\}$ to be the set of numeric values in the input, where x can be the review quantity n or user rating $(5.0 - r)$. We use $(5.0 - r)$ as the formula to represent user rating since developers are typically more interested with topics with lower ratings. The following paragraphs describe how we extract the various insights. Three of the insights (salient topics, abnormal topics, and correlated topics) are extracted following a recently proposed data mining method in [8].

Salient Topics: Salient topics are the topics that occupy significantly large proportions or significantly low user ratings in current versions. We hypothesize that the values in X follow a power-law distribution with Gaussian noises. After sorting X in the descending order and removing the maximum value $\{x_{max}\}$, we fit the remaining values, *i.e.*, $X \backslash \{x_{max}\}$, to a power-law distribution (shown in Figure 2 (a)) and use the prediction errors of $x_i \in X \backslash \{x_{max}\}$ to approximate the Gaussian distribution $N(\mu, \sigma^2)$. With the power-law distribution, we obtain the predicted error of the maximum x_{max} by $\varepsilon_{max} = \hat{x}_{max} - x_{max}$. Then based on the Gaussian distribution (shown in Figure 2 (b)), we calculate the probability to achieve the predicted error ε_{max} by $p = Pr(\varepsilon > \varepsilon_{max} | N(\mu, \sigma^2))$. Finally, we compute the significance of the maximum x_{max} as $\text{Sig}_{\mathcal{S}} = 1 - p$.

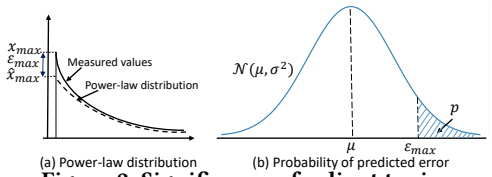


Figure 2: Significance of salient topics.

Abnormal Topics: Abnormal topics are the topics that exhibit significant increases in review numbers or significant drops in user ratings when compared with previous versions. Let w be the window size, which is the number of previous versions to analyze. For example, $w = 5$ means that whether the topics are anomalies in current versions are determined by the previous five versions. Given the observed values in a windows size $X_w = (x_{v-w}, ..., x_{v-1}, x_v)$, where v denotes the current version, we first fit X_w to a line by linear regression and compute its slope *slope* and the goodness-of-fit [8] value r^2. Similarly, we obtain the slope of each topic in version v. We then use Gaussian distribution $N(\mu, \sigma^2)$ to model the distributions of slopes, based on which we compute the probability of the slope value s equal to or larger than the observed slope by $p = Pr(s > |slope| | N(\mu, \sigma^2))$. Finally, we define the significance of each slope as $\text{Sig}_{\mathcal{A}} = r^2 \cdot (1 - p)$.

Causal Factors: Causal factors are extracted when the overall ratings or total review numbers for the current app version are significantly different than the previous version. The aim of capturing such insights is to dig out the factors that dominate the changes.

Let $X = (x_1, x_2, ..., x_t)$ be the time series of measured values (*e.g.*, overall ratings or total review numbers) of an app. We first use the convolution-based moving average method [1] to detect the significant values in the time sequence, as shown in Figure 3 (a). The moving average is predicted based on discrete linear convolution,

i.e., $avg_n = \sum_{i=1}^{t} x_i * g_{n-i}$, where $g \in \mathbb{R}^t$, $g = (1/w, 1/w, ..., 1/w)$, and w is the predefined window size. Then we calculate the standard deviation σ of residuals between the moving averages and observations, *i.e.*, $(x - avg) \sim N(0, \sigma^2)$. Finally, version v is considered to have significant change if $||x_v - avg_v|| > \sigma$, as illustrated in Figure 3 (b). The causal factor $\mathcal{Y}$ is determined as the topic with the largest increase rate in review quantity (or the largest decrease rate in user rating).

Correlated Topics: Correlated topics are the topics that exhibit strong correlations with respect to their review quantity or user ratings. Extracting such topics can help developers discover the topics that may impact certain app issues. We use X_1 and X_2 to denote the measured values of two topics in consecutive versions. We first compute the Pearson correlation between these two sequences $\rho(X_1, X_2)$ and obtain the p-value. Then we hypothesize that the correlation coefficient follows the Normal distribution $N(\mu, \sigma^2)$ [8], where $\mu = 0$, $\sigma = 0.5$. Thus, the significance $\text{Sig}_{\mathcal{E}}$ of the correlation of two topics can be calculated by $Pr(\rho(X_1, X_2)|N(\mu, \sigma^2))$.

2.2 Text Template Definition

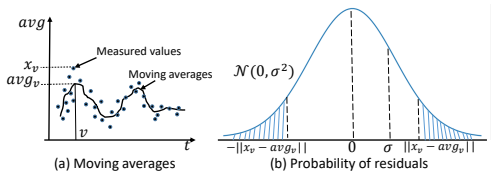
The descriptions of insight summary should not only cover the insight types, but also explain the importance to developers in a comprehensible manner. Therefore, the insights generated by INFAR is designed to contain both the important topics and the corresponding reasons. In this work, we define a few text templates specific to each insight type. For example, we have "*Version ... has abnormal topic ... in review number, which shows the review number increases ... compared with the last version.*" for describing the abnormal topics.

2.3 User Review Prioritization

When retrieving user reviews relevant to specific topic or word, INFAR will display the reviews ranked by their importance. We use the similar prioritization method in [4], which involves two aspects, *i.e.*, review length h and user rating r. The importance score of one review is defined as $exp(-r/\log(h))$, which means that reviews with longer lengths and lower ratings will be ranked higher.

3 HOW TO USE INFAR

INFAR is web-based application that analyzes raw user reviews and generates an insight summary of the reviews. INFAR takes the output of SURF as its input, SURF [3] will parse the input data, predict the topics of each review The raw reviews in the input file is saved as "*[rating]******[review text]******[post date]******[version]*" per line, using "******" to space these review attributes. One example review from YouTube of iOS is "*1.0******unable to restart, delete or download again. what's up?******Mar 20, 2016******11.07*". We define such input format due to its simplicity. After uploading the raw



(a) Moving averages (b) Probability of residuals

Figure 3: Significance of causal factors.

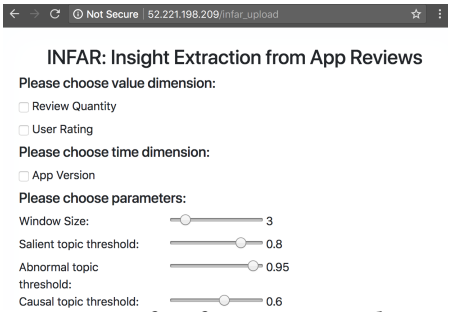


Figure 4: Interface for parameter selection.

reviews through the entry page, the server side of INFAR will parse the input data, predict the topics of each review using SURF [3], prioritize the reviews with the method described in Section 2.3, and save the processed reviews into app version SQLite database. INFAR automatically identifies the app version of these reviews correspond to and provides the interface for parameter selection, depicted in Fig. 4. The parameters include window size w, significance threshold for salient topic, significance threshold for abnormal topic, and threshold for causal factor σ. With all parameters set, INFAR can then generate insight summary based on the techniques described in Section 2.

The insight summary includes two large parts, one for word distributions in user reviews and another for extracted insights[2]. Fig. 5 illustrates the word distributions using word cloud, where the font size indicates the word importance (*i.e.*, tf-idf value[3]) and the font color denotes which type of reviews the word generally appears in. Users can click specific words to see the related prioritized reviews. The review lists also support basic retrieval, such as filtering reviews of specified types and topics. The salient topic is explained with pie chart for review quantity or bar chart for user rating. By clicking one topic, users can also observe corresponding prioritized reviews. All the other topics are described with line charts for users to observe the trends of these topics along versions.

4 CASE STUDY

To evaluate the usefulness of the insight summary generated by INFAR, we have conducted empirical evaluation involving six popular apps publicly available and 12 staff in several large IT companies. The six subject apps have been utilized in our previous work [4]. They are distributed in two different app stores (i.e., Google Play and App Store) and span across the app stores' six different categories. These app received a total of 164,026 reviews that we collected from August 2016 to April 2017. For the 12 participants, 50% of them have over four years of engineering experience, and only two of them have fewer than one year of engineering experience. They are one product manager, two testing engineers, four development engineers, two researchers, and one intern. For evaluating INFAR, we (i) prepared the input for each subject app; (ii) randomly assigned the input file of an app to each participant for practicing INFAR; and (iii) invited participants to answer questions of one short survey. The demo tool website depicts the questions of the survey, with aggregated data regarding the answers provided by participants.

[2]Note that there would be 0~4 types of insights for each version.
[3]https://en.wikipedia.org/wiki/Tf-idf

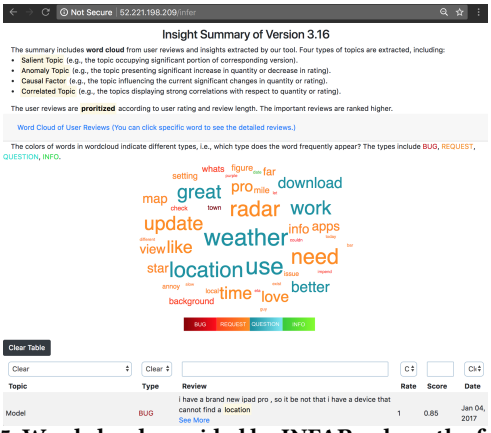


Figure 5: Word cloud provided by INFAR, where the font size indicates the word frequency and the color denotes which type of reviews the word usually appear. By clicking some word, users will see the prioritized relevant reviews below.

The majority of participants 92% (11/12) agreed that whole provided insights useful, with 42% (5/12) of them judging these insights to be highly useful. Also, the majority of them 58.3% (7/12) totally approved that the insight summaries are comprehensible, while the rest of them partially agreed with that. All the participants said that analyzing user reviews is very hard or hard without INFAR. Moreover, 6/12 of subjects declared that INFAR summaries allow them to save at least 50% of the time as compared to manually analyzing user reviews. Regarding the quality of the extracted insights, 4/12 of them said that the insights do not contain any unnecessary information, while the rest of them stated that they contain partially unnecessary information. By looking deeply at the usefulness of each insight type, 11/12, 11/12, 10/12, 8/12, and 10/12 agreed with the usefulness of word cloud, salient topics, abnormal topics, causal factors, and correlated topics, respectively. Also, over 50% of participants approved that all these displayed insights are highly useful for them except for causal factors with 4/12 approval. Overall, INFAR is affirmed to be useful for app development and save the total time for analyzing user reviews.

5 RELATED WORK

INFAR is mainly related to tools for assisting user review analysis. We briefly describe the tools below. MARK [9] is one tool that supports keyword-based search. MARK retrieves keywords similar to the query keyword and lists the most relevant reviews. ARMiner [2] and PAID [5] are techniques for prioritizing app reviews. The closest tool to INFAR, SURF [3, 7], is a popular tool for app review classification built by Di Sorbo *et al.* Given a set of reviews, SURF can visualize the number of reviews that fall under different topics as interactive bar charts. Although their tool can assist developers in identifying topics of interest (*e.g.*, the ones involving more reviews), the tool cannot produce issue topic comparison from multiple time periods.

Different from INFAR, the above-mentioned studies are unable to produce contrastive summaries capturing the four insight types considered in this work. The closest work to INFAR are MARK [9] and SURF [3, 7]. Different from INFAR, MARK cannot capture review topics and produce contrastive summaries in terms of these topics. Also, different from INFAR, SURF cannot compare topics across multiple time periods. SURF also does not produce natural language summaries.

Table 1: Example Insight of YouTube.

Insight Type	Example Insight
Salient Topic	Version 11.13 has explainable topic (Feature/Functionality) with review number 233, which accounts for significant proportion 49.8% in that version.
Abnormal Topic	Version 3.16 has Abnormal topics (GUI,Contents, Feature/Functionality) in review number, which shows the review number increases 125.00%,81.82%,55.56% compared with the last version.
Causal Factor	Version 11.15 experiences significant increase in the review number by 7.7%. This is mainly attributed to topics (GUI), which show an increase of 31.6% compared with the last version.
Correlated Topic	Version 11.17 observes two topics (App,Model) that present strong correlations in review number, with maximum correlation values at 0.97.

6 CONCLUSION

In this work, we create one tool for extracting four kinds of important insights that capture contrastive information summarized from app reviews for developers. Our empirical evaluation shows that INFAR is promising in helping developers efficiently compare user reviews across time periods and analysis dimensions. In future, we are interested in extending INFAR to capture more insight types and in evaluating INFAR through a more comprehensive user study involving more industrial participants.

ACKNOWLEDGMENT

The work was fully supported by Microsoft Research Asia (2018 Microsoft Research Asia Collaborative Research Award), the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14210717 of the General Research Fund), and the National Natural Science Foundation of China (No. 61332010, 61472338).

REFERENCES

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.
[2] Ning Chen et al. "AR-Miner: mining informative reviews for developers from mobile app marketplace". In: *Proceedings of the 36th International Conference on Software Engineering (ICSE)*. ACM. 2014, pp. 767–778.
[3] Andrea Di Sorbo et al. "What would users change in my app? summarizing app reviews for recommending software changes". In: *Proceedings of the 24th SIGSOFT International Symposium on Foundations of Software Engineering (FSE)*. ACM. 2016, pp. 499–510.
[4] Cuiyun Gao et al. "Online App Review Analysis for Identifying Emerging Issues". In: *Proceedings of the 40th International Conference on Software Engineering (ICSE)*. ACM. 2018.
[5] Cuiyun Gao et al. "Paid: Prioritizing app issues for developers by tracking user reviews over versions". In: *Proceedings of the 26th International Symposium on on Software Reliability Engineering (ISSRE)*. IEEE. 2015, pp. 35–45.
[6] Xiaodong Gu and Sunghun Kim. "" What Parts of Your Apps are Loved by Users?"(T)". In: *Proceedings of the 30th International Conference on Automated Software Engineering (ASE)*. IEEE. 2015, pp. 760–770.
[7] Andrea Di Sorbo et al. "SURF: summarizer of user reviews feedback". In: *Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, May 20-28, 2017 - Companion Volume*. 2017, pp. 55–58.
[8] Bo Tang et al. "Extracting Top-K Insights from Multi-dimensional Data". In: *Proceedings of the 2017 International Conference on Management of Data (SIGMOD)*. ACM. 2017, pp. 1509–1524.
[9] Phong Minh Vu et al. "Mining user opinions in mobile app reviews: A keyword-based approach (t)". In: *Proceedings of the 30th International Conference on Automated Software Engineering (ASE)*. IEEE. 2015, pp. 749–759.