TOUR: Dynamic Topic and Sentiment Analysis of User Reviews for Assisting App Release

Tianyi Yang
The Chinese University of Hong Kong
Hong Kong, China
tyyang@cse.cuhk.edu.hk

Cuiyun Gao*
Harbin Institute of Technology
Shenzhen, China
gaocuiyun@hit.edu.cn

Jingya Zang
Harbin Institute of Technology
Shenzhen, China
zjyzangjingya@163.com

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

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

ABSTRACT

App reviews deliver user opinions and emerging issues (e.g., new bugs) about the app releases. Due to the dynamic nature of app reviews, topics and sentiment of the reviews would change along with app release versions. Although several studies have focused on summarizing user opinions by analyzing user sentiment towards app features, no practical tool is released. The large quantity of reviews and noise words also necessitates an automated tool for monitoring user reviews. In this paper, we introduce **TOUR** for dynamic **TO**pic and sentiment analysis of **U**ser **R**eviews. TOUR is able to (i) detect and summarize emerging app issues over app versions, (ii) identify user sentiment towards app features, and (iii) prioritize important user reviews for facilitating developers' examination. The core techniques of TOUR include the online topic modeling approach and sentiment prediction strategy. TOUR provides entries for developers to customize the hyper-parameters and the results are presented in an interactive way. We evaluate TOUR by conducting a developer survey that involves 15 developers, and all of them confirm the practical usefulness of the recommended feature changes by TOUR.

CCS CONCEPTS

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

KEYWORDS

App review, review topic, sentiment analysis

ACM Reference Format:
Tianyi Yang, Cuiyun Gao, Jingya Zang, David Lo, and Michael R. Lyu. 2021. TOUR: Dynamic Topic and Sentiment Analysis of User Reviews for Assisting App Release. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3442442.3458612

*Cuiyun Gao is the corresponding author.

WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8313-4/21/04.
https://doi.org/10.1145/3442442.3458612

1 MOTIVATION

App user reviews reflect users' instant experience with the app. Developers are eager to know users' opinions about their apps after release, including which features are favorable or unfavorable by users, presence of bugs, and desired new requirements. Timely distilling user reviews could facilitate the app release process for developers, e.g., addressing issues or adding new features in the next release. For example, Facebook Messenger received massive low ratings in August 2014 and suffered a great loss of users, because the version contained serious privacy issues (e.g., access to users' mobile phone photos and contact numbers) [11]. However, the complaints about the privacy issue by a few users had already surfaced right after the release on Apple's App Store. The serious issue could be effectively mitigated if it were timely identified from user reviews.

The large volume and noisy characteristics of user reviews [2] increase the burden of manually checking the reviews version by version. Existing commercial platforms for app market analytics, such as App Annie [1], simply list all user reviews without providing in-depth analysis of user reviews. Prior studies on automatic review analysis either rely on manually annotated data for training [5], which is labor-intensive, or directly adopt common sentiment analysis tools for predicting the user opinions [6, 8]. However, according to Novielli et al.'s study [9], the common sentiment analysis tools are proven not to perform well for software engineering tasks. Moreover, they do not consider the dynamic nature of mobile apps, i.e., different app versions are associated with different reviews.

Based on our survey of developers (see Section 5), the vast majority said getting a detailed analysis of user reviews is important. Developers demand a tool that can automatically detect the issues and user sentiment about app features along with app releases.

2 KEY INNOVATIONS

In this paper, we demonstrate TOUR, a customizable and automatic tool for dynamic **TO**pic and sentiment analysis of **U**ser **R**eviews. TOUR helps app developers automatically track the topic changes and user sentiment about app features along with versions. With an app version chosen, TOUR provides a "glimpse" for each topic of user reviews in both phrase and sentence levels, and highlights the emerging topics, so that developers can focus on the important

[1] https://www.appannie.com/

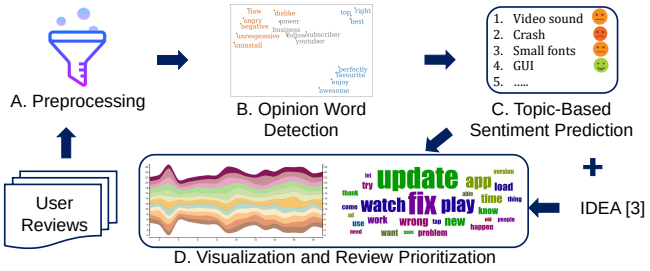


Figure 1: Workflow of TOUR.

ones. TOUR also presents the results of the sentiment analysis in an interactive mode. The key innovation of TOUR is an approach for lightweight sentiment analysis of emerging issues in app user reviews based on customizable opinion words instead of external sentiment tools.

3 WORKFLOW OF TOUR

The workflow of TOUR mainly includes four steps, i.e., preprocessing, opinion word detection, topic-based emerging issue detection and sentiment prediction, and visualization and review prioritization, as shown in Figure 1.

3.1 Preprocessing

The preprocessing step aims at formatting raw data and removing meaningless reviews for subsequent analysis. Following previous work on app review analysis [4], we first remove all non-English characters and all non-alphanumeric symbols except the punctuation and then conduct stemming with Porter's Stemmer to convert each word to its root form. We further clean the review texts based on rules defined in [4], such as correcting consecutive duplicates (e.g., *"suuuuper"* is converted into *"super"*), removing consecutively duplicate words (e.g., "*very very annoying*" is converted into "*very annoying*"), and removing all the words whose length is more than 15 (since the word lengths of 99.95% English words are less than 16[2]). The cleaned reviews are fed into the next step.

3.2 Opinion Word Detection

In this step, we aim at identifying opinion words and their sentiment polarities, i.e., negative, neutral, or positive. We identify opinion words by aspect words which usually describe app features and tend to be nouns [5]. For the review "*I like Facebook App's multimedia features but the battery consumption sucks*", the aspect words are "*multimedia*" and "*battery consumption*". We then extract opinion words according to the relations with the aspect words, where the relations are captured according to the semantic dependency graph [1] of the review sentence. To accurately extract the opinion words, we choose three types of dependency relations, i.e., *noun of subject*, *direct object*, and *adjective modifier*, as listed in Table 1. The opinion words are identified if they present one type of the dependency relations with the aspect words. Figure 3 illustrates an example of the semantic dependency graph for the review text "*it is so slow and it glitches up*". We can observe that the opinion

[2]http://norvig.com/mayzner.html

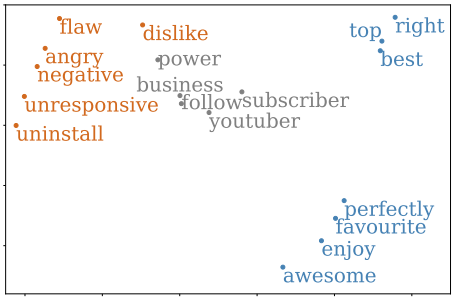


Figure 2: Visualization of word embeddings after dimensionality reduction with t-SNE [12]. The positive opinion words are colored blue and the negative opinion words are colored orange. The aspect words are colored gray.

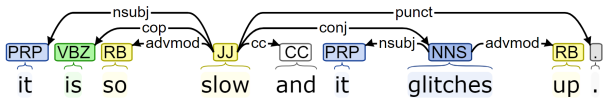


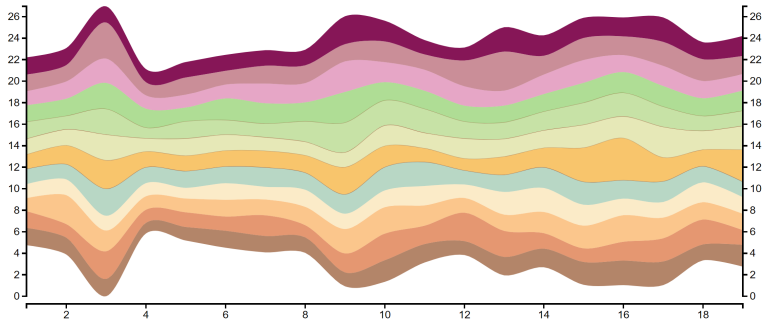
Figure 3: An example of parsed semantic dependency graph.

Table 1: Aspect words and opinion words in different contexts.

Dependency	Examples of **aspect words** and *opinion words*
noun of subject	This **app** *crashed* on launch.
direct object	I *dislike* the **app**.
adjective modifier	Book the *cheapest* **flight**.

words, i.e., "*slow*" and "*glitches*", present an `nsubj` (indicating normal subject) relationship with the corresponding aspect word, i.e., "*it*".

To predict the sentiment polarities of the opinion words, prior studies [5, 6] rely on common sentiment analysis tools that have been shown [9] to be inaccurate for software engineer datasets. Besides, for different apps, the same opinion words can also exhibit totally different sentiment polarities. For example, for the review of YouTube, "*Can you not make the ad **louder** than the music videos?*", the opinion word "*loud*" conveys negative sentiment; while for the review of NOAA Radar "*I really like the feature of sending a **loud** alert to my phone during pending dangerous weather*", the word "*loud*" is positive in sentiment. To adaptively estimate the sentiment of opinion words, we retrain fine-tuned existing word embeddings [10] with every app's reviews. To validate the effectiveness of the retrained embeddings, we conduct dimension reduction on the retrained word embeddings of YouTube with t-SNE [12] and visualize them in Figure 2. The positive opinion words are colored blue and the negative opinion words are colored orange. The aspect words are colored gray. Figure 2 shows that opinion words expressing positive and negative sentiment are separated well. We then adopt the seed polarity words released by Wilson et al. [14] as the base seed word sets for sentiment prediction. The seed word sets contain manually-labeled context-free polarity words, e.g., "*great*" and "*hate*". Note that developers can enrich the seed words by manually entering domain-specific polarity words. The



(a)

Version	12.07
Label	wont load; cellular data; stupid message
Sentences	1. this be important for me and i can't sta nd now be able to watch my favourite youtube channel; 2. i be a dig fan of youtube and also upload every now and then and i can't now because it wont load it say in offline when i can clearly see i'm online; 3. i'm an app developer for io and this app have a terrible ux;

(b)

Figure 4: (a) Visualization of issue river. X-axis and Y-axis indicate the version and the width of issue river, respectively. All the topics constitute the issue river, and each branch of the river represents one topic/issue highlighted by a different color. (b) Issue with representative sentences and sentence labels.

sentiment polarities of the opinion words for each app are finally predicted based on the cosine distances with the seed words. Specifically, we define the positive seed word set as $\mathcal{P}$ and the negative one as $\mathcal{N}$. For each opinion word w, the sentiment $S(w)$ is defined as:

$$S(w) = \sum_{n \in \mathcal{N}} \frac{Sim(w,n)}{|\mathcal{N}|} - \sum_{p \in \mathcal{P}} \frac{Sim(w,p)}{|\mathcal{P}|}, \quad (1)$$

where $S(w) \in [-1, 1]$ and larger sentiment polarities indicate more negative opinions. $Sim(w,p)$ or $Sim(w,n)$ present the cosine distances between the opinion word w and the positive seed word $p \in \mathcal{P}$ or negative seed word $n \in \mathcal{N}$, respectively.

3.3 Topic-Based Emerging Issue Detection and Sentiment Prediction

TOUR tracks the emerging issues with our previous work, IDEA [3], where emerging issues are detected by version and the topics are automatically labeled. It utilizes user reviews as input and outputs labeled topics for each version. It employs an online topic modeling approach to generate version-sensitive topic distributions. The emerging topics are then identified based on a typical anomaly detection method. To make the topics comprehensible, IDEA labels each topic with the most representative phrases and sentences. TOUR further predicts the sentiment of each labeled topics. Each topic $t \in T$, where T denotes the set of all extracted topics, is represented by a set of topic words $\{w_1^t, w_2^t, \cdots\}$ ranked by the probability distributions. For ensuring the semantic representativeness of the topic words, we consider the top 30 words of each topic during analyzing the topic sentiment. Specifically, TOUR first assigns a sentiment label l to each top word w, where the sentiment label is defined according to the sentiment score $S(w)$, i.e., $l \in L$, and L = {Strongly Positive, Positive, Weakly Positive, Slightly Positive, Slightly Negative, Weakly Negative, Negative, Strongly Negative}. Based on the assigned labels, TOUR directly adopts the sentiment label associated with the most top words as the sentiment label of the topic.

3.4 Visualization and Review Prioritization

TOUR provides an interactive interface for assisting developers monitoring the topic changes and user sentiment towards app features along with app versions. The topic changes are visualized through issue river [3] and user sentiment analysis results are illustrated as word cloud. TOUR also prioritizes the reviews for each topic to facilitate developers' further analysis. The prioritization is based on the probability distributions of the reviews under the topic, with the corresponding topic words highlighted.

4 SYSTEM DEMONSTRATION

TOUR is a web application that monitors the topic changes and user sentiment about app features along with app release, where emerging app issues are alerted and reviews of each topic are prioritized for each app release. An interactive demonstration of TOUR will be provided, and the developers can interact with TOUR. The following scenarios will be demonstrated.

4.1 Parameter Selection

The input of TOUR is a text file where each line is organized as *"[rating]******[review text]******[post date]******[version]******[region]", using "******"* to space these review attributes. One example review from YouTube of iOS is *"1.0******Pls fix this. The last update fails to load and play video.******Mar 29, 2017******12.11******SG"*. As shown in Figure 5, after uploading reviews, developers will set model parameters including the number of topics, the threshold of the topic probability for prioritizing reviews of each topic, and other parameters of IDEA such as the number of previous versions to be considered for modeling the topics of current version. TOUR provides a comprehensive list of base seed words and developers can enrich seed words with their domain-specific polarity words.

4.2 Evolution of Topics

The evolution of topics along with versions will be exposed to the developers through a user interface. The issue river [3] is employed to display topic variations. By moving the mouse on one branch, i.e., one topic, in the issue river, the developers can track the topic

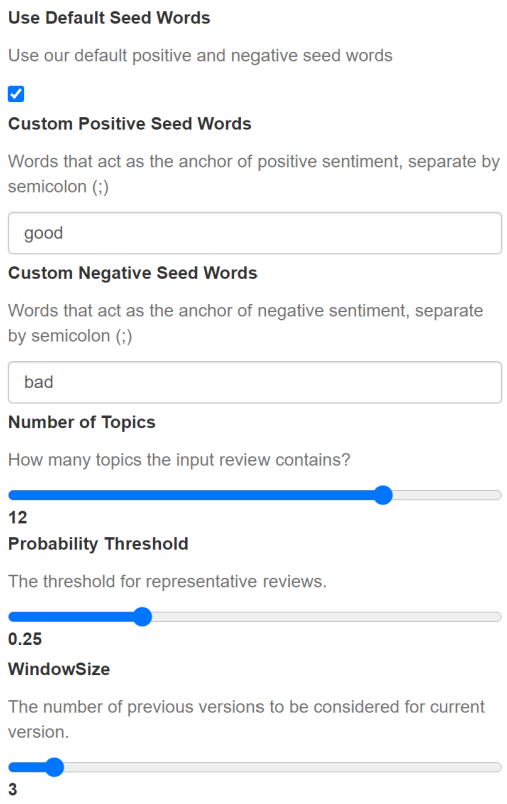


Figure 5: The interface for parameter selection.

changes along with app versions. Figure 4(a) presents an example of the visualized issue river for YouTube iOS. All the app issues constitute the issue river and each branch of the river indicates one topic, highlighted in a different color. The topics with wider branches are of greater concern to users. The width of the k-th branch in the t-th version is defined as: $\text{width}_k^t = \sum_a \log Count(a) \times Score_{sen}(a)$, where $Count(a)$ is the count of the phrase label a in the review collection of the t-th version, and $Score_{sen}(a)$ denotes the sentiment score of the label a.

4.3 Glimpses of Topics

When clicking on one branch, the topic "glimpse" will automatically appear, as shown in Figure 4(b), including representative sentences and sentence labels. For the emerging topics/issues, the sentences are highlighted in yellow for reminding developers. One can also view user sentiment about the topic, visualized in the form of a word cloud, as illustrated in Figure 6. Larger font sizes indicate that the word presents higher probability distribution in the topic and the topic words are displayed in different colors, and the colors indicate the sentiment of the topic words.

4.4 Prioritized Reviews

Prioritized reviews associated with each topic, as illustrated in Figure 7. In the ranked review list, review texts and corresponding attributes such as rating, post date, app version, and relevance score

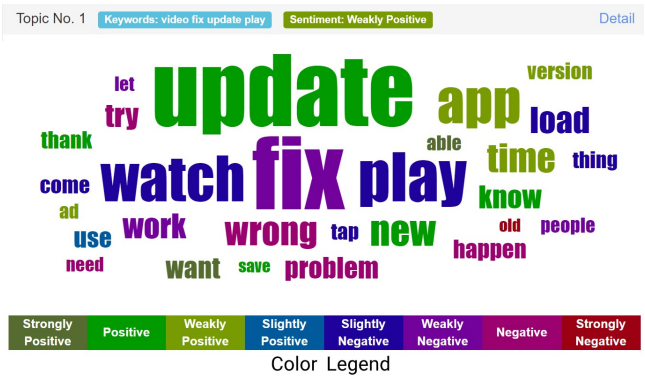


Figure 6: Visualization of issue-sentiment summarization. Font size in the word cloud denotes probability weight of the word in the topic and the color indicates the sentiment label of the word.

(i.e., topic probability distribution) are also displayed for developer's reference. To illustrate users' sentiment in review text, the topic words are highlighted with the same colors as those in the word cloud. The review lists also support full-text retrieval. By typing keywords in the search bar, developers can filter reviews by content, rating, and date, and digest user reviews in more details.

5 CASE STUDY

In order to evaluate the usefulness of topic-sentiment summary generated by TOUR, we conducted an empirical evaluation involving 15 developers. For the 15 participants, 73% of them have over one year of software development experience, and 29% of them have over three years of software development experience. We adopted the review repository of the six popular apps released by Gao et al. [3], including YouTube, NOAA Radar, Clean Master, eBay, Swift Keyboard, and Viber. Specifically, we (i) prepared the input reviews for each app; (ii) randomly chose one app and showed the generated issue river, word cloud, and representative reviews to participants; and (iii) invited participants to answer several questions about the usefulness and clarity of the results.

Our GitHub page[3] depicts the questions and the statistics regarding the answers from the participants. Overall, 14 out of the 15 participants considered topic summarization and sentiment analysis of app reviews "important" or "very important", while only one of them insisted that the task was "not that important". All participants agreed that the provided analysis results are useful as a whole, with 46% of them considered highly useful. Also, the majority of them (13/15) agreed that the visualization are totally comprehensible, while the rest of them partially agreed with that. Most of the participants (14/15) said it is hard to analyze user reviews without TOUR. Additionally, nearly half of them (7/15) declared that our tool saves them at least 50% of the time, compared to manually analyzing the user reviews. Additionally, 73% of them declared that our tool save them over 30% of the time, compared to manually analyzing user reviews.

[3]https://yttty.github.io/tour

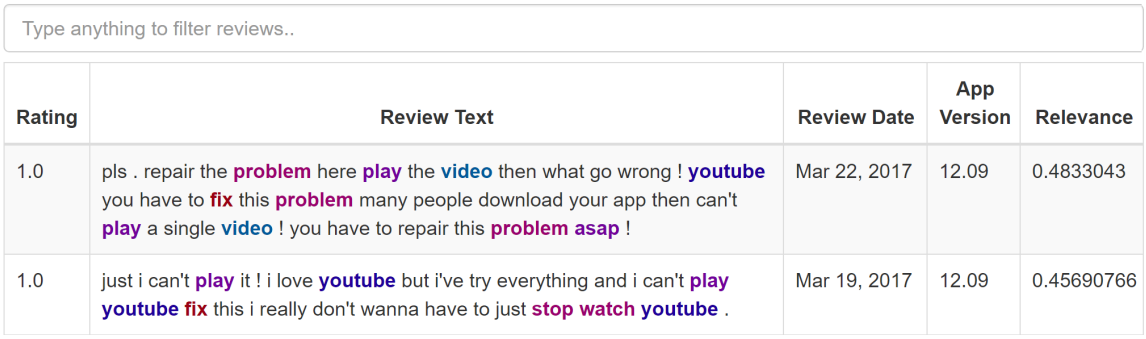


Figure 7: Prioritized associated reviews provided by TOUR. The topic words are colorized to indicate the sentiment of the topic words.

Regarding the quality of the results, all of them said that the topic-sentiment summary contains necessary information for review analysis. Besides, 14/15, 15/15, and 15/15 agreed with the usefulness of word cloud visualization, sentiment analysis, and prioritized representative reviews, respectively. "We pay great attention to user experience and feedback. Your tool is quite helpful.", a senior developer commented. To sum up, TOUR is helpful for developers and provides useful assistance for user review understanding.

6 RELATED WORK

Existing works [5, 7, 8, 13] mainly aim at automatic retrieval of app feature requests from reviews. They mainly analyze static reviews and pay little attention to tracking issue changes. Gu and Kim [5] identify reviews related to aspect evaluation based on manually-annotated reviews, and then use the associated reviews for aspect-opinion analysis. Luiz et al. [8] also adopts topic modeling to identify app features, but does not consider the emerging issue detection, customizable sentiment seed words, parameter adjustment, and review prioritization involved in our tool.

7 CONCLUSION

App user reviews are valuable for developers but are difficult to analyze. In this work, we develop an online tool for monitoring topic changes of app reviews along with app release and capturing user sentiment towards app features and issues. An empirical evaluation shows the effectiveness of TOUR for assisting developers in efficiently analyzing user reviews. In future, we will conduct a more comprehensive empirical evaluation with more developers.

ACKNOWLEDGMENTS

The work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14210920), and the National Natural Science Foundation of China under project No. 62002084.

REFERENCES

[1] Daniel Cer, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning. 2010. Parsing to Stanford Dependencies: Trade-offs between speed and accuracy. In *7th International Conference on Language Resources and Evaluation (LREC 2010)*. http://nlp.stanford.edu/pubs/lrecstanforddeps_final_final.pdf

[2] Andrea Di Sorbo, Sebastiano Panichella, Carol V Alexandru, Junji Shimagaki, Corrado A Visaggio, Gerardo Canfora, and Harald C Gall. 2016. 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*. ACM, 499–510.

[3] Cuiyun Gao, Jichuan Zeng, Michael R. Lyu, and Irwin King. 2018. Online app review analysis for identifying emerging issues. In *Proceedings of the 40th International Conference on Software Engineering, ICSE 2018*. 48–58.

[4] Cuiyun Gao, Wujie Zheng, Yuetang Deng, David Lo, Jichuan Zeng, Michael R. Lyu, and Irwin King. 2019. Emerging app issue identification from user feedback: experience on WeChat. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE (SEIP) 2019, Montreal, QC, Canada, May 25-31, 2019*, Helen Sharp and Mike Whalen (Eds.). IEEE / ACM, 279–288.

[5] Xiaodong Gu and Sunghun Kim. 2015. What Parts of Your Apps are Loved by Users?. In *30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015*. 760–770.

[6] Emitza Guzman and Walid Maalej. 2014. How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews. In *IEEE 22nd International Requirements Engineering Conference, RE 2014, Karlskrona, Sweden, August 25-29, 2014*. 153–162.

[7] Claudia Iacob, Varsha Veerappa, and Rachel Harrison. 2013. What are you complaining about?: a study of online reviews of mobile applications. In *BCS-HCI '13 Proceedings of the 27th International BCS Human Computer Interaction Conference, Brunel University, London, UK, 9-13 September 2013*. 29.

[8] Washington Luiz, Felipe Viegas, Rafael Odon de Alencar, Fernando Mourão, Thiago Salles, Dárlinton B. F. Carvalho, Marcos André Gonçalves, and Leonardo C. da Rocha. 2018. A Feature-Oriented Sentiment Rating for Mobile App Reviews. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*. ACM, 1909–1918.

[9] Nicole Novielli, Daniela Girardi, and Filippo Lanubile. 2018. A benchmark study on sentiment analysis for software engineering research. In *Proceedings of the 15th International Conference on Mining Software Repositories, MSR 2018, Gothenburg, Sweden, May 28-29, 2018*, Andy Zaidman, Yasutaka Kamei, and Emily Hill (Eds.). ACM, 364–375.

[10] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[11] Dave Smith. 2014. Facebook Messenger is Getting Slammed by Tons of Negative Reviews Right Now. http://www.businessinsider.com/facebook-messenger-app-store-reviews-are-humiliating-2014-8 Accessed: 2021-03-24.

[12] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[13] Phong Minh Vu, Tam The Nguyen, Hung Viet Pham, and Tung Thanh Nguyen. 2015. Mining User Opinions in Mobile App Reviews: A Keyword-Based Approach (T). In *30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015*. 749–759.

[14] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. *Computational Linguistics* 35, 3 (2009), 399–433.