**Table 1: Intention Categories Definition**

| | |
|---|---|
| **Information Giving** | Sentences that inform other users or developers about some aspect of the app. |
| **Information Seeking** | Sentences describing attempts to obtain information or help from other users or developers. |
| **Feature Request** | Sentences expressing ideas, suggestions or needs for enhancing the app. |
| **Problem Discovery** | Sentences reporting unexpected behavior or issues. |
| **Other** | Sentences not belonging to any of the previous categories. |

a publicly available data collection containing such a substantial amount of data, in which reviews related to specific app releases are labeled according to software maintenance categories (i.e., types of user feedback) and apps are analyzed by static analysis tools for computing their software quality.

## 2 RELATED WORK

Despite app stores represent a relatively recent phenomenon, they immediately captured the interest of the software engineering community and, nowadays, there are already over 180 papers devoted to their study[9]. As a consequence, several datasets involving a quite high numbers of apps with structured (e.g., source code) and unstructured information (e.g., commits messages) have been proposed in the literature. For instance, the paper by Krutz *et al.*[8] provided a dataset that reports results obtained by several static analysis tools on 4,416 different versions of 1,179 open-source android applications combined with data of version control commits related to these applications. Collections containing huge amounts of app reviews have also been published for pursuing different research goals. For example, the *Data Set for Mobile App Retrieval*[3] includes 1,385,607 user reviews of 43,041 mobile apps and it has been mainly used to run experiments about accuracy improvements in mobile app retrieval[15]. The *SoftWare Marketplace* (SWM) review dataset[4] contains 1,132,373 reviews from 15,094 apps and has been involved in research works aimed at detecting spam or fake reviews[2, 18, 19]. Other existing public available data[5] could be used to build and test sentiment analysis algorithms, since they contain reviews clustered according to the sentiment expressed in them (i.e., negative and positive sentiment). Nevertheless, to the best of our knowledge, no previous work provided a comprehensive dataset that, at the same time, (i) sheds the light on the types of feedback users report for different versions of several apps and, (ii) combines such information with software quality indicators computed on the app versions they are referring to.

## 3 DATASET CONSTRUCTION

Our dataset was built in two phases: (i) in the data collection phase we analyzed the F-Droid repository and the Google Play store for collecting the app versions data and the information related to their user reviews; (ii) in the analysis phase we examined the Android package (i.e., the apk) of the mined apps using several static analysis scripts/tools and labeled the extracted reviews through the use of two automated classifiers.

[3]https://sites.google.com/site/daehpark/Resources/data-set-for-mobile-app-retrieval
[4]http://odds.cs.stonybrook.edu/swmreview-dataset/
[5]https://github.com/amitt001/Android-App-Reviews-Dataset

**Table 2: Topic Definitions**

| Cluster | Description |
|---|---|
| **App** | sentences related to the entire app, *e.g.*, generic crash reports, ratings, or general feedback |
| **GUI** | sentences related to the Graphical User Interface or the look and feel of the app |
| **Contents** | sentences related to the content of the app |
| **Pricing** | sentences related to app pricing |
| **Feature or Functionality** | sentences related to specific features or functionality of the app |
| **Improvement** | sentences related to explicit enhancement requests |
| **Updates/ Versions** | sentences related to specific versions or the update process of the app |
| **Resources** | sentences dealing with device resources such as battery consumption, storage, etc. |
| **Security** | sentences related to the security of the app or to personal data privacy |
| **Download** | sentences containing feedback about the app download |
| **Model** | sentences reporting feedback about specific devices or OS versions |
| **Company** | sentences containing feedback related to the company/team which develops the app |
| **Other** | sentences not treating any of the previous topics |

### 3.1 Data Collection Phase

In this phase, we primarily built a web crawler (available in the dataset URL) to collect from the F-Droid repository the meta-data (package name, available versions, release date of each version) and the apks of each app. The crawler initially mined data for 1,929 different apps. The versions of each mobile application have been ordered according to the release date (i.e., from the oldest to the latest version). All the apps (i) not appearing in the Google Play Store and (ii) whose latest version was released before the year 2014 (*i.e.*, this could indicate that the app is no longer maintained) have been discarded. A second scraper tool[6] was built to download from Google Play Store all the user reviews related to the remaining 965 apps. It relies on Phantom JS[7] and Selenium[8] in order to navigate the Play Store web site and extract reviews from the resulting HTML code. We set up a *cronjob* in order to mine new reviews 4 times a week. The tool totally gathered 297,323 app reviews, and for each user comment it also extracted (i) the package name of the app to which the review refers, (ii) the review content, (iii) the related star-rating assigned by the user to the app, and (iv) the posting date of the review. Relying on the release date of each applications' version and on the review's posting date of each user comment, we assigned each review to one of the app versions as described below. Given a generic version of an app, $V_i$, and the next version of the same app, $V_{i+1}$, the reviews assigned to the version $V_i$, i.e., $R_i$, are collected considering the reviews whose posting date occur after the release date of $V_i$ and before the release date of $V_{i+1}$. Despite this assumption may produce for some reviews an assignment to a wrong app version, Pagano et Maalej [10] empirically demonstrated that user feedback is mostly triggered by new releases, i.e., usually in the first few days after the download of a new app version. We discarded 8,758 reviews (because their publication date was too old for assigning them to any of the available versions) obtaining a dataset containing 288,565 reviews belonging to 710 different versions. Then we decided to keep in the collection exclusively the app versions having at least 10 reviews assigned (according to previous studies [17]), discarding all the remaining ones. At the end of this filtering process we obtained a dataset of 288,065 reviews related to 629 versions of 395 different apps.

[6]https://github.com/sealuzh/user_quality/tree/master/tools
[7]http://phantomjs.org/
[8]http://www.seleniumhq.org/