Create Dataset for Sentiment Analysis by Scraping Google Play App Reviews using Python

Learn how to scrape reviews for Android apps and use the information to build a dataset for sentiment analysis

Venelin Valkov Follow
May 2 · 4 min read ★

Photo by Markus Winkler

> TL;DR Learn how to create a dataset for Sentiment Analysis by scraping user reviews for Android apps. You'll convert the app and review information into Data Frames and save that to CSV files.

- Run the notebook in your browser (Google Colab)
- Complete project on GitHub

You'll learn how to:

- Set a goal and inclusion criteria for your dataset
- Get real-world user reviews by scraping Google Play
- Use Pandas to convert and save the dataset into CSV files

Setup

Let's install the required packages and set up the imports:

The Goal of the Dataset

You want to get feedback for your app. Both negative and positive are good. But the negative one can reveal critical features that are missing or downtime of your service (when it is much more frequent).

Lucky for us, Google Play has plenty of apps, reviews, and scores. We can scrape app info and reviews using the google-play-scraper package.

You can choose plenty of apps to analyze. But different app categories contain different audiences, domain-specific quirks, and more. We'll start simple.

We want apps that have been around some time, so opinion is collected organically. We want to mitigate advertising strategies as much as possible. Apps are constantly being updated, so the time of the review is an important factor.

Ideally, you would want to collect every possible review and work with that. However, in the real world data is often limited (too large, inaccessible, etc). So, we'll do the best we can.

Let's choose some apps that fit the criteria from the *Productivity* category. We'll use AppAnnie to select some of the top US apps:

Scraping App Information

Let's scrape the info for each app:

We got the info for all 15 apps. Let's write a helper function that prints JSON objects a bit better:

Here is a sample app information from the list:

This contains lots of information including the number of ratings, number of reviews and number of ratings for each score (1 to 5). Let's ignore all of that and have a look at their beautiful icons:

App icons from our dataset

We'll store the app information for later by converting the JSON objects into a Pandas dataframe and saving the result into a CSV file:

Scraping App Reviews

In an ideal world, we would get all the reviews. But there are lots of them and we're scraping the data. That wouldn't be very polite. What should we do?

We want:

- Balanced dataset — roughly the same number of reviews for each score (1–5)
- A representative sample of the reviews for each app

We can satisfy the first requirement by using the scraping package option to filter the review score. For the second, we'll sort the reviews by their helpfulness, which are the reviews that Google Play thinks are most important. Just in case, we'll get a subset from the newest, too:

Note that we're adding the app id and sort order to each review. Here's an example for one:

`repliedAt` and `replyContent` contain the developer's response to the review. Of course, they can be missing.

How many app reviews did we get?

```
15750
```

Let's save the reviews to a CSV file:

Summary

Well done! You now have a dataset with more than 15k user reviews from 15 productivity apps. Of course, you can go crazy and get much much more.

- Run the notebook in your browser (Google Colab)
- Complete project on GitHub

You learned how to:

- Set goals and expectations for your dataset
- Scrape Google Play app information
- Scrape user reviews for Google Play apps
- Save the dataset to CSV files

Next, we're going to use the reviews for sentiment analysis with BERT. But first, we'll have to do some text preprocessing!

References

- Google Play Scraper for Python

. . .

Originally published at https://www.curiousily.com.