



Sentimend | Google Play Sentiment Explorer



Google Play



Gemini



GitHub

Sam Graler and Randy Hucker

Outline

- Introduction
 - Motivation
 - What is Sentimend?
- How Does it Work?
 - Pipeline/Tech Stack
 - Demo!
- Evaluation
 - Experiment Design
 - Results
- Conclusion



Introduction

Motivation

- App reviews represents a huge source of actionable feedback for developers, but they are difficult use directly of for several reasons:
 - There is a large amount of data to review
 - The data is unstructured
 - Developers often have more pressing concerns
- Sentiment analysis tools that could aid developers in this process are expensive and generally focused on managing public relations/meeting marketing objectives



Sentimend allows developers to tap into this underutilized source of feedback by providing *development-focused recommendations* specifically designed to address problems extracted from user reviews (all while remaining *responsive* and *low-cost*)

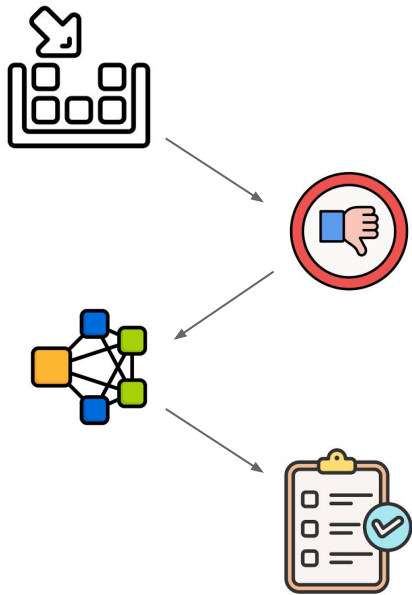


What is Sentimend?

Sentimend is a free, open-source tool that helps developers extract actionable engineering insights from large volumes of real user reviews without using expensive SaaS products or manual triage.

What Sentimend Does:

- Collects live reviews from Google Play
- Detects negative sentiment
- Clusters similar issues using transformer-based embeddings
- Summarizes the core issue in each cluster and proposes actionable fixes using LLMs
- Creates GitHub issues to track the identified issues





How Does it Work?

Pipeline/Tech Stack

Pipeline Overview:

1. Accepts an app name or URL from the Google Play Store
2. Automatically scrapes recent user reviews for the selected app
3. Performs sentiment analysis to isolate reviews that are most likely to provide insights
4. Clusters related complaints using sentence embeddings and unsupervised clustering
5. For each cluster, uses an LLM to summarize the unifying theme among the reviews and suggests a potential feature or enhancement addressing those user frustrations
6. Optionally integrates with GitHub to generate new issues for the proposed solutions



All results are displayed in a streamlined UI designed to allow developers to operate as efficiently as possible

Pipeline/Tech Stack

Tech Stack Description:

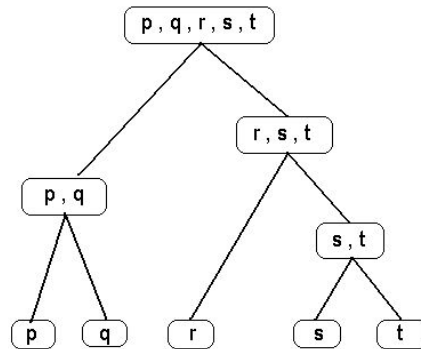
1. Accepts an app name or URL from the Google Play Store
2. Automatically scrapes recent user reviews for the selected app
 - a. Live review acquisition via google-play-scraper
3. Performs sentiment analysis to isolate reviews that are most likely to provide insights
 - a. VADER (compound score -1 to 1) for fast polarity detection
 - b. Filters to negative reviews only (high-signal)
 - c. Supports swapping in a different or even custom sentiment analyzer for customizability



Pipeline/Tech Stack

Tech Stack Description:

4. Clusters related complaints using sentence embeddings and unsupervised clustering
 - a. Hierarchical Agglomerative Clustering - each review starts as a cluster, clusters are then merged bottom-up based on proximity until we have larger meaningful groups
 - b. Ward's method with Euclidean distance - when combining groups, we choose the merge that keeps the group as tight (minimize distance) and focused (minimize variance via Ward's method) as possible
 - c. Auto or fixed-K cluster selection



Pipeline/Tech Stack

Tech Stack Description:

5. For each cluster, uses an LLM to summarize the unifying theme among the reviews and suggest a potential feature or enhancement addressing that user frustration
 - a. Summarizes cluster themes
 - b. Suggests actionable feature ideas
 - c. Drafts GitHub issues complete with acceptance criteria
6. Optionally integrates with GitHub to generate new issues for the proposed solutions
 - a. Users may select which issues (if any) from the above step they'd like to be automatically created on GitHub



Demo!

<http://localhost:8501>

Evaluation



Experiment Design

Goal: Evaluate how well Sentimend helps developers perform the following tasks:

- Analyze the sentiment of user reviews
- Cluster complaints coherently such that they properly correspond to user frustrations

Procedure

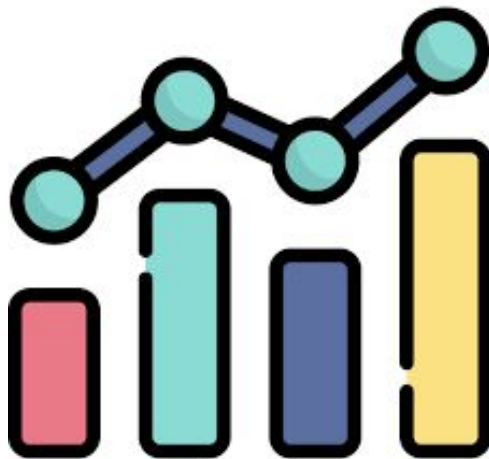
- Apps: Spotify (500), Google (1000), WhatsApp (2000)
- Manual vs. Automated analysis on same datasets
- For each dataset:
 - Conduct manual sentiment analysis & clustering
 - Run Sentimend pipeline
 - Compare cluster quality, coverage, effort reduction



Experiment Design

Metrics

1. Cluster Coherence & Quality: Silhouette score + spot-checks
2. Coverage of User Problems: Manual issues vs. clusters + summaries
3. Developer-Oriented Actionability: How useful are the recommended solutions/generated issues
4. Effort Reduction: How much time was saved by Sentimend?



Results

Quantitative Findings

- Cluster Quality:
 - Silhouette score ≈ 0.04
 - Modest, but reasonable for noisy, unstructured app reviews
- Effort Reduction:
 - Negative filtering reduces reviews 48–60%
 - Sentiment analysis performed for 2500 reviews in <3 seconds
 - Clustering performed for 2500 reviews in <5 seconds:
 - With a manual clustering time of 25-45 minutes for 200 reviews, that represents a x3750 speed up at minimum for the clustering task

Results

Qualitative Findings

- Cluster Quality:
 - Clusters reflect clear themes: crashes, login pain, intrusive ads, performance issues, account locks
 - Although they sometimes overlap, the themes are consistent within the clusters themselves
- Coverage of User Problems:
 - Major problems in user reviews were consistently identified by Sentimend (sometimes redundantly)
- Developer-oriented actionability:
 - LLM issue drafts are usable starting points for engineering tasks, but they lack application specific technical context that would be useful for a practical issue

Conclusion

What Sentimend can do:

- Turn thousands of raw reviews into clear, actionable insights
- Extract real user pain points in seconds to minutes instead of minutes to hours to days
- Generate practical engineering tasks ready for GitHub

Why it matters:

- Reviews are a massive untapped source of product direction
- Sentimend can help teams listen to users at scale
- Open-source design enables low-cost experimentation and integration

What would be next to improve (future work):

- More comprehensive evaluation of Sentimend's capabilities compared to other paid tools
- Enhance the quality of LLM generated issues (perhaps using repository context?)

Thank You!

Questions...?