

# Analyzing Public Discussions for Product Insights

Mining Reddit and Tiki for product issues, sentiment, and trends

From noisy threads → actionable product signals.

[Placeholder image: collage of anonymized comment snippets with callouts]

## **Problem & Value**

- Which products break, why, and how often—using public discussions.
- Turn unstructured Reddit and Tiki text into actionable product signals.
- Output insights teams can ship on: issues, trends, severity, examples.

# Objectives

- Which issues? How frequent? How trending? What severity?
- Where do issues cluster (by product/subreddit/time)?
- How reliable are methods vs baselines?
- Deliver a repeatable pipeline and clear, prioritized findings.

## Why These Platforms?

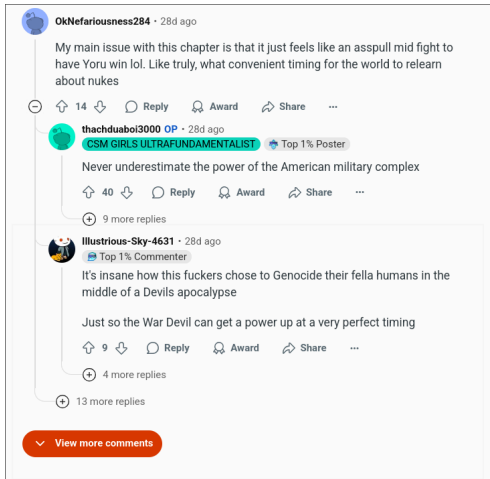
- Reddit: rich, threaded discussions; public API; strong NLP support.
- Tiki (e-commerce): structured, purchase-verified reviews; VN market signal.
- Complementary: text-rich threads vs short reviews → broader coverage.
- [Placeholder image: two-column pros/cons cards + coverage map]

# Data Collection

- **Initial:** PRAW (Python Reddit API Wrapper) for prototyping.
  - **Limitation:** 1,000 post cap, rate limits.
- **Current (Hybrid):**
  - Reddit historical archives (Academic Torrents) to bypass API caps.
  - Tiki review dumps for cross-source validation.
  - **Result:** Broader time windows, more volume.
- [Placeholder image: timeline ribbon PRAW → Archives → Tiki]
- [Placeholder image: flowchart with rate-limit icon on PRAW; time-filter lock]

# How PRAW Traverses Comments

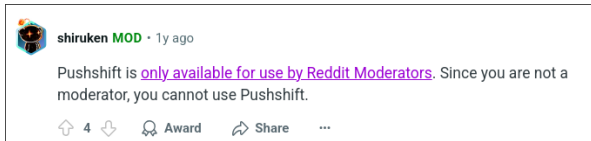
- Reddit returns comment trees with View more comments placeholders.
- We expand lazily, traverse, and serialize —no one-click “download”.
- Captures full depth where needed; avoids rate-limit explosions.



# Alternatives Considered

## Pushshift.io (Pushshift API)

- More powerful than PRAW, can filter posts by time
- Requires moderator status  
→ not feasible



## Personal Archive


- Continuous collection over weeks
- Impractical: hardware/time, can't capture older posts

(Or building one's own archive over a long period of time like the OP mentioned in another comment, that works too – but it does take time. Though they could load it with data from these archives too if they were so inclined.)

## Academic Torrents (Arctic Shift)

- Downloadable historical Reddit datasets
- Good for history + scale

### Download tool



Download posts and comments from a subreddit or user. Very large subreddits can take a long time to download. In that case, you can maybe narrow down the time range. Alternatively, you can download [subreddit dumps through Academic Torrents](#) or [monthly dumps](#).

r/

u/

headphones

Approximately 396k posts and 4.29m comments

Start date

2010-02-26

End date

now

☒ Download posts

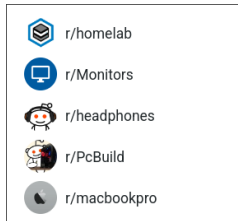
☒ Download comments

Start



# Subreddits Chosen

- A diverse mix of tech and lifestyle communities:
- r/macbookpro, r/GamingLaptops
- r/iphone, r/AppleWatch, r/Monitors, r/headphones, r/homelab, r/photography
- ...and several others covering home, audio, and PC building.
- total : 134121 posts, 1300190 comments, 2025-06-01 -> 2025-07-31.



```
prawl_tests/subreddits/blank_slate on / main [x!?] via 🐙
) : polars open all_posts_hanni.parquet | polars shape
+-----+
| # | rows | columns |
+-----+
| 0 | 134121 | 45 |
+-----+

prawl_tests/subreddits/blank_slate on / main [x!?] via 🐙
) : polars open all_comments_hanni.parquet | polars shape
+-----+
| # | rows | columns |
+-----+
| 0 | 1300190 | 25 |
+-----+
```

## EDA Highlights

- [Placeholder image: time series of comments/week by subreddit]
- [Placeholder image: length distribution (boxplot or histogram)]
- [Placeholder image: top product mentions (bar) via keyword/NER]
- Takeaways (placeholder): spikes follow launch X; r/homelab comments 2× longer.

# Preprocessing Pipeline

Convert 32 JSONL  $\rightarrow$  2 Parquet (Nushell + Polars); keeping as many meaningful columns as possible.

## Before

Posts: **106** columns

Comments: **69** columns



## After

Posts: **28** columns

Comments: **17** columns

- **Cleaning:**

- Removed URLs.
- Filtered for English-only content for initial analysis. (only 5 thousands comments)

- Randomized 100 posts/sub (seeded) to reduce virality bias. (placeholder)
- [Placeholder image: before/after text snippet (URLs/emojis removed)]
- [Placeholder image: language donut (kept vs dropped)]
- [Placeholder image: schema alignment diagram JSONL → typed Parquet]

## Context Experiment (Per-Comment SA)

- Parent-context vs per-comment classification compared on small labeled set.
- Result: per-comment avoids 512-token truncation and context pollution.
- [Placeholder image: ablation bar chart (F1 with vs without parent context)]

## Sentiment Analysis (SA)

- **Approach:** Pretrained transformers over simpler baselines (VADER, TextBlob).
- **Model:** lxyuan/distilbert-base-multilingual-cased-sentiments-student
  - **Why:** Good multilingual support, lightweight, strong zero-shot performance.
- **Execution:** Rented GPU on `vast.ai` for large-scale inference.
- **Next Step:** Fine-tune on a domain-specific labeled dataset.
- [Placeholder image: model compare bar (VADER/TextBlob/DistilBERT) on labeled set]
- [Placeholder image: throughput/cost bar CPU vs GPU on vast.ai]
- [Placeholder image: confusion matrix or calibration curve]
- [Placeholder panel: 2–3 qualitative examples with predicted labels]

## Topic Modeling (Planned)

- **Goal:** Discover key themes and issues per subreddit.
- **Method:** Evaluate BERTopic vs. traditional methods (LDA/NMF).
- **Process:** Preprocess with domain stopwords, lemmatization.
- **Output:** Top topics, representative comments, and trend lines.
- Target: coherence c\_v threshold and stable topics across subsamples. (placeholder)
- [Placeholder image: intertopic distance map (BERTopic mock)]
- [Placeholder image: top-words table for 2 sample topics]

## CLI Tool (Planned)

- A pipeline for repeatable analysis:
- ingest: Raw data to Parquet.
- clean: Filter and normalize data.
- sentiment: Run batch sentiment analysis.
- topics: Train and apply topic models.
- report: Aggregate results and export.



## Risks & Mitigations

- **Bias:** Sampled randomly across multiple, diverse subreddits.
- **Time Drift:** Included time slices to compare cohorts.
- **Reproducibility:** Pinned environments, config-driven runs, and stored seeds.
- **Ethics:** Used public data, followed platform ToS, aggregated results.

## Next Steps

- Finalize EDA visualizations.
- Run sentiment analysis at scale.
- Pilot and select a topic modeling approach.
- Build and demo the core CLI workflow.
- Begin labeling for fine-tuning sentiment model.

## Q & A