

Forum: [Hack Forums](#)

Thread: [My journey building a tool for doomscrolling awareness + Facebook post analysis](#)

Oh_Well - October 9, 2025, 2:21 pm

My journey building a chrome extension for
doomscrolling awareness + Facebook post analysis

Hey guys, as my 1st entry to this awesome weekly contest by Quantum

<https://hackforums.net/showthread.php?tid=6310391>

I wanted to share a personal project this time, that I started recently.

A period passed when I wasn't good, I had dark thoughts and the algorithms weren't helping. I found myself scrolling endlessly on negative-vibed feeds for hours, until I decided that I didn't want to do that anymore.

So I build a Chrome extension that tackles doomscrolling while also experimenting with Facebook post analysis.

Social platforms are designed for engagement, not for our well-being.

I wanted to build a tool that does three things for me:

1. Raise awareness while scrolling - Giving me gently reminders when I'm slipping into doomscrolling patterns.
2. Analyze the content itself - run sentiment and thematic analysis on facebook, so I get insights into what kind of content is influencing my mood.
3. Block harmful/negative content - During the 2nd process, it's analyzing posts I didn't even see yet on my scroll view, so the harmful content was blurred out and it had a message about the type of content and an option to view it anyways.

Chrome extension base - hooks into the DOM while browsing.

Client-side analysis - I've been exploring lightweight ML models running in-browser (privacy-preserving, no server round trips).

Hybrid methods - in some cases, offloading analysis to a backend for deeper NLP tasks, but keeping the sensitive data on the client whenever possible.

Sentiment & topic analysis - the extension flags highly negative posts, patterns of repetitive themes, and gives the user a snapshot of what they've consumed in the last X minutes.

Frontend (Extension):Manifest V3 (Chrome's latest extension format)

Content scripts injected into Facebook pages

Popup UI built with vanilla JS and CSS (keeping it lightweight)

Chrome storage API for persisting user preferences and analytics data

Analysis engine: TensorFlow.js for in-browser ML inference
Sentiment analysis using a stripped-down BERT model (quantized to ~20MB)
Layer of keyword extraction using TF-IDF implementation
Real-time DOM mutation observers to catch new posts as they load

Backend - which is completely optional if you want to make it lighter: Node.js + Express for advanced analysis
PostgreSQL for storing anonymized aggregate data
Redis for caching frequent analysis results
OpenAI API integration for deeper content analysis - still optional - you can only use Tensorflow if u want

When you first install the extension, it walks you through a quick setup:
Set your "scroll time threshold" (default: 15 minutes)
Choose notification style (subtle, moderate, or assertive)
Select which topics to track (politics, news, entertainment, etc.)
Privacy mode: client-only vs. hybrid analysis

As you browse Facebook, the extension silently works in the background:

What it does: Tracks scroll depth and time spent on page
Analyzes each post you see (text content, not images yet)
Builds a real-time sentiment profile
Detects repetitive negative patterns

What you see:
A small toolbar that changes the highlighted color based on overall sentiment
Green = mostly positive content
Yellow = mixed or neutral
Red = predominantly negative content

Hidden posts / images if the content is too negative - based on your onboarding selection.

When you hit your threshold (say, 15 minutes of scrolling with 70% negative content), you get a notification:
You can dismiss it, snooze for 10 minutes, or view your detailed dashboard.

Challenge 1: Facebook's DOM structure

Spoiler ([Click to View](#))

Facebook uses heavily obfuscated class names and dynamic rendering. Posts don't have consistent selectors, so I had to:

- Build a heuristic system that identifies posts based on DOM structure patterns

- Use multiple fallback selectors

- Implement a "learning mode" that adapts to FB's changes

- Monitor mutation observers for dynamically loaded content

Challenge 2: Sentiment analysis accuracy

Spoiler ([Click to View](#))

Training data is biased, and social media language is weird:

Problems:

- Sarcasm breaks most models ("Oh great, another election post" is negative despite "great")

- Emoji analysis (? vs ?)

- Mixed languages in posts

- Context matters (a negative post about a good outcome: "Can't believe this corruption finally ended")

My solutions:

- Using emoji sentiment dictionaries as a supplementary signal

- Implementing context windows (analyzing 2-3 sentences together)

- Letting users flag incorrect analyses to improve the model

- Keeping confidence scores and only showing high-confidence results

- Using LLMs to make more accurate predictions

Challenge 3: Performance

Spoiler ([Click to View](#))

Running ML models in-browser is slow:

Initial problems:

- Extension slowed down Facebook noticeably

- Battery drain on laptop

- Memory leaks from keeping too much data

Optimizations I made:

- Lazy loading: only analyze posts in viewport + 2 screens ahead

- Web Workers: offload analysis to background threads

- Model quantization: reduced model size from 85MB to 22MB

- Debouncing: batch analyze multiple posts together

- Caching: don't re-analyze posts you've already scrolled past

What I've learned

Technical lessons: Web scraping is fragile - always have fallbacks
In-browser ML is viable but requires heavy optimization
Performance debugging is 50% of the work

Personal lessons:

Testing my own extension changed my scrolling habits
Side projects don't have to be perfect to be valuable

Resources & References

If you want to build something similar, here are resources that helped me:
Chrome Extension docs (Manifest V3)
TensorFlow.js documentation
Hugging Face model hub (for pre-trained models)
DistilBERT for lightweight sentiment analysis
OpenAI API for further analysis / better predictions

I'd love to get your feedback and if it's something you want to work on it, I'll prep the code and open source it

PS: "Quantum" styled text images were made by the awesome tool [CL1B](#) created

<https://hackforums.net/showthread.php?tid=6310489>