# Comment-Toxicity-Meter

With this browser extension I want to continue my previous exploration of sentiment analysis and use it to change YouTube to reflect their detected toxicity.

This extension is meant to provoke in several ways, first it can give a brief overview of the comments under a video without necessarily having to read them. This means a user of this extension could quickly by scrolling through the comments recognize a color pattern under each video and then decide whether they want to deal with these comments. Reading overly negative and toxic comments can be mentally taxing, and this tool gives the user a warning, and could influence their decision of further engaging with these comments.

On the other hand, sentiment analysis is not perfect, yet it is used to assess people, and their engagement in content. But even the most sophisticated sentiment analyses have their weak spots, their false positives/negatives. This is an (un)intended side-effect of the interaction. The user of the browser extension gets to see how the algorithm (naïve algorithm based on assigning words and emojis with negative and positive numbers, depending how strongly positive or negative the words where considered, and calculating a score based of that.) rates different comments and make their own judgement whether this comment was fairly judged. This comments on machines ability to recognize human intent, and the ethical questions involved when being judged by an obviously flawed (but potentially powerful) algorithm.

# Steps and Implementation

I first attempted to build an add “enhancer” for facebook, but that proved more difficult than anticipated: <https://www.vice.com/en/article/7xydvx/facebooks-arms-race-with-adblockers-continues-to-escalate>

Then I tried to build one for youtube, but I could only catch the banners on the page but not the ad-videos themselves.

So, I decided to use the comments under the YouTube videos and display them differently depending on the sentiment they are projecting.

This extension runs in chrome and injects and analyzes content from youtube.com.

Implementation. I use the sentiment npm package to perform the sentiment analysis on the text. To get the npm packages required working in a chrome extension, I bundled them using browserify. Npm packages typically require a server/client setup to run, but this allows me to bundle up the scripts I need and run them on the client side. When a site is loaded I check if this site is youtube.com and then the extension starts searching for comments in the DOM structure of the youtube page repeatedly (to update the page while scrolling) and color the comments depending on the sentiment they project.

# Usage:

To install the browser extension download this repository. The Youtube\_node.js script is the script that needs a node.js server to run and load the node-packages.

Node Variant:

To install the node packages run npm install inside the folder.

Change the youtube\_node.js file as you wish, then use browserify to bundle the npm packages into a single file. For browser extensions, required js libraries must be bundled this way, otherwise they will not function.

Non Node:

Alternatively use the youtube.js script, which includes the already code from the used npm packages bundled into a single file using browserify.

Set the script to be loaded within the youtubeScript() function in the background.js file

# Next Steps:

Use extension to filter censor comments or filter them.

Expand to other sites such as Facebook or Twitter

More intelligent Sentiment analysis (maybe one that recognizes political affiliation or irony or one that tries to identify the “strongest” comments in the comments)