# Detecting and Mitigating Toxic Comments in Online Platforms Using Toxic BERT and Topic Modeling

By

Mervin McDougall

Marina Mitiaeva

Lakshmi Peram

Nick DeVita​

Group #1

## Project Overview

This project aims to enhance Reddit's moderation process and foster a healthier environment by introducing a **proactive system** **to detect high-risk topics that are likely to attract viral toxic comments**. Our study will analyze a dataset of questions and top-voted answers to (1) identify highly engaging topics using metrics like upvote scores and (2) assess the toxicity of comments within those threads using the Toxic-BERT model, evaluating the relationship between toxicity and engagement. In the future, this model could be deployed for **real-time analysis**, enabling moderators to efficiently respond to harmful content and promote safer community interactions.

The problem we address is **business-oriented**, focusing on improving the quality and sustainability of Reddit’s engagement. Current moderation efforts on Reddit rely on a multi-layered approach that includes **automated tools, human review, and user reports**. While Reddit claims to be "constantly building and refining tools to proactively identify content and behavior that violates policies," (Reddit)  its **use of NLP remains limited, with notable efforts in harassment detection using** **large language models (LLMs)** only recently emerging before the company’s IPO​ (The Register). Despite these advancements, two-thirds of the moderation process remains manual, placing significant pressure on volunteer moderators. Additionally, Reddit’s reliance on unpaid moderators has made it vulnerable to disruptions, as evidenced by the 2023 protests when moderators opposed new data access fees for third-party app developers.​ (CNA).

Our approach introduces a **novel, preventive solution** by combining **topic modeling with toxicity detection**, enabling early detection of viral discussions with toxic elements. Unlike reactive measures such as user reports or keyword-based filtering, our system will **proactively identify high-risk conversations** with both high engagement and toxicity. By integrating this solution with Reddit’s current moderation tools, we aim to **reduce manual effort, enhance efficiency, and improve moderation workflows**.

The key stakeholders are **Reddit users**, whose well-being depends on positive interactions; **moderators**, who require better tools to manage content efficiently; **shareholders**, given the company’s recent IPO, with a vested interest in platform sustainability; and **advertisers**, whose brand safety relies on a healthy, well-moderated platform environment.

## Prediction Inference and Other Goals

The main goal throughout the project was to find the topics of content on Reddit’s threads that amount to toxicity. Toxic content has always been apparent on the internet and does not show signs of stopping anytime soon, and the entertainment it brings to a subset of users cannot be overlooked. Flagging anything deemed toxic or offensive and taking it down from any platform is not an equitable or sustainable solution but having a clearer idea of the topics that generate toxicity, and pugnacious discourse would benefit Reddit’s users, advertisers, and moderators.

## Data Exploration

The dataset consisted of three main features – the title, body, and answers. The title and body are exactly as the feature names describe, the title and body of the thread. The answers, however, consist of an array of dictionaries that include properties for the answer body and score. The total dataset contained 1,833,556 observations. However, due to memory restrictions, we could only import and work with 50,000 observations.

Initially, our approach to data exploration saw the body feature and answers feature as two separate elements. Consequently, we first investigated the body, passing it through tokenization, filtering, frequency distribution, and then word cloud display.

To address the filtering of stop words, we employed the stop word lists from three libraries – Spacy, NLTK, and SciKit Learn. We combined all three lists to filter out the stop words. We also provided a custom list of stop words which we procured by performing multiple frequency distributions.

The word cloud developed from the body feature identified words such as “know”, “time”, “people”, “thing”, “going”, “want” and “friend” were the most prominent. Among the amalgamation of all the words for the body, only 3% of the words were unique.

Figure 1

Given the close relationship between the answers and the body, we decided to properly understand and identify the topics we needed to join the features into what we refer to as a thread. **A thread is the concatenation of the Title, body, and all answers related to that observation.**

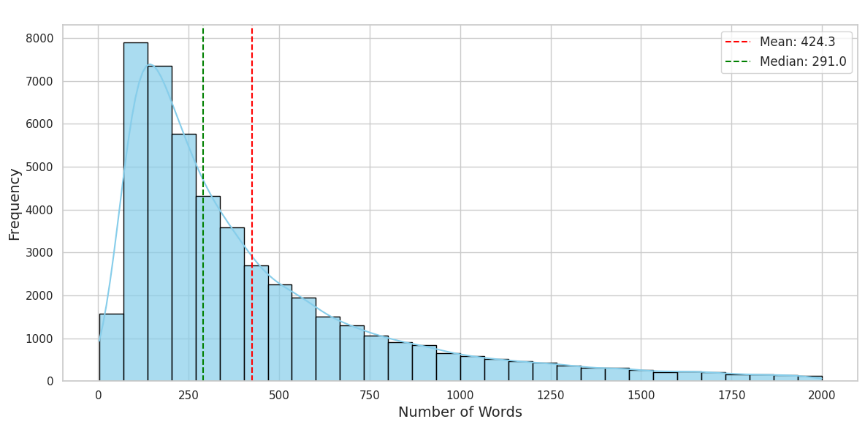
Having gone through the same process of filtering, on the corpora of threads, we identified some interesting statistics about the corpora. The smallest thread was 3 words whereas the longest was 97,756 words. The median number of words was about 291 words with a mean of 424 words. This suggests a skew to the right for the distribution of threads. The 25th percentile had 162 words, the 50th percentile had 302 words and the 75th percentile had 592 words.

Figure 2

## Unexpected Results

One of the most surprising results came from the relationship between the popularity of a thread and the level of toxicity in that thread. It was initially thought there was a strong positive correlation between the toxicity of a thread topic and its popularity. Our tests on the Reddit data showed just the opposite. There was a negative correlation between the popularity of the thread topic and the level of toxicity as depicted in this regression plot.

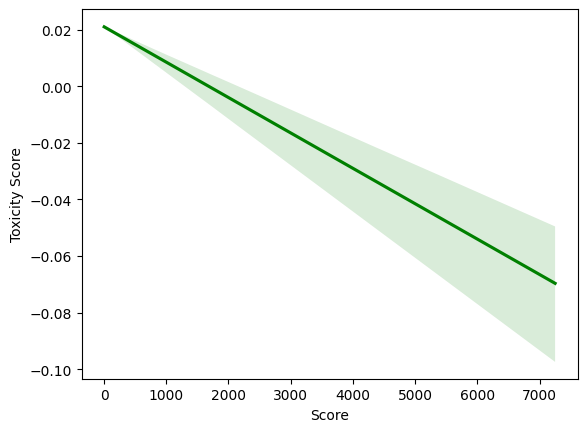


Figure 3

Another unexpected result was the relationship between topics and topic words generated through the topic modeling process. The reason for creating the thread structure indicated in the Data Exploration section was to encapsulate an entire topic in one structure. Therefore, through topic modeling, we could analyze whether there was a drift between the topic and the responses in the threads. We were incorrect in that assumption. The topic title and the topics generated from the threads were on par with each other.

## Summary of Methods Used

The dataset was retrieved from Hugging Face and contained a listing of Reddit questions and answers (HuggingFace). The dataset consisted of four features- the title, the body, the score, and the answers. The title body and answers were combined into one corpus we refer to as a thread. The threads were the data structure that was eventually used for processing. To examine what methods were used to analyze and process the dataset, we need to split the project into some component stages preprocessing, EDA, and finally development.

**Preprocessing:**

The dataset was limited to 50,000 rows as opposed to the complete 1,833, 556 rows of the dataset. This was due to memory restrictions on the chosen platform for development – Google Colab. To prepare the data for further processing, stop words were filtered from the body and the answers. To accomplish this, stop word lists from multiple libraries were merged and used to filter out the most common stop words. The libraries used included NLTK, SciKit Learn, and Spacy. Tokenization of the data was done through the NLTK library. Regular expression filtering was performed to remove non-alphabetic stop words and numbers from the corpora. In the preprocessing steps, frequency distribution was used to identify stop words for more granular control over the stop word filtering process.

**EDA:**

Exploratory Data Analysis was facilitated through visualization libraries such as MatplotLib, Seaborn, and Wordcloud. This allowed us to display statistical information about the corpora to identify patterns in the data. Some of the visualizations used included frequency distributions, thread length, and word clouds. Frequency distribution was again used in the EDA process to provide a clearer view of the most common words in the corpora.

**Development:**

Topic modeling was achieved using LDA (Latent Dirichlet Allocation) and was performed on the generated threads of the observations. Topics were limited to the top 20 topics. Throughout all stages, PySpark was utilized to manage the data being processed and analyzed through EDA. However, in the development process, the data was extracted from PySpark into a basic Pandas data frame for further processing as it was discovered there was an incompatibility between PySpark, and the Toxic Bert library provided by Hugging Face. (Hanu).

## Results Summary

**Topic Modeling:**

First, the topic words were like those used in the title of the threads. This indicated an alignment between the generated topics and the overall guiding themes of the individual threads. In short, the responses to the topic did not go off-topic.

Secondly, titles that shared similar topics often discussed similar themes. For example, threads that had topics like “government”, and “party” were focused on political themes. This also hinted at repetition among threads.

Lastly, when the topics were aligned with the corresponding score for the threads, the high-scoring posts reflected popular, general themes rather than unique ones.

**Toxic BERT Scoring:**

Contrary to prior beliefs, we discovered there was a negative correlation between the popularity of the thread and the accompanying toxicity score generated by Toxic Bert. With the data we were able to extract, we identified the correlation between Score and Toxicity Score was -**0.0732** where the Score represented the popularity of the topic, and the Toxicity Score represented the value generated by Toxic Bert.

The relationship was not a smooth negative slope. The level of toxicity undulated a lot with lower scores and became less as the scores increased. Consequently, the higher the popularity of the topic, the less toxicity was discovered in the threads.

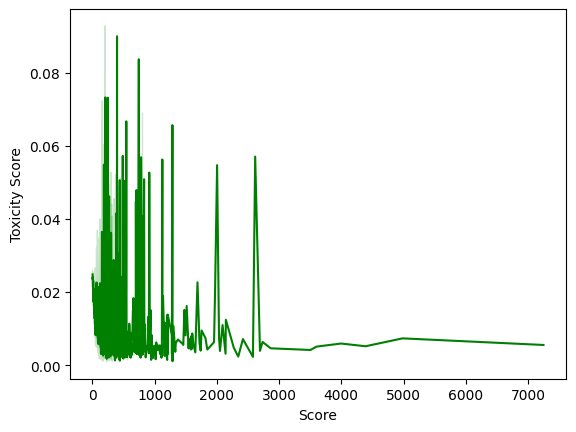


Figure 4

## Problems Encountered

The problems encountered with the project can be grouped into four main categories. These included memory restrictions, selective stop-word filtering, incompatibilities with libraries, and the use of multiple libraries. A more detailed description of the problems encountered, and the associated fixes are described in the following paragraphs.

Memory restrictions were first encountered when downloading the dataset from hugging face for processing. The dataset consisted of 1.8 million observations, and this exceeded the memory restrictions of the free tier for Google Colab. To address this, we limited the records to 50,000 for our analysis and continued development.

Another memory restriction was noticed while performing the Exploratory Data Analysis (EDA). Because of the immutable nature of PySpark variables, many intermediate data structures like data frames continued to linger in memory once processing on them was complete. This created issues with memory management as the processing of the project continued to run out of memory. For variables that were no longer needed, their values were set to None as soon as they were no longer required to free up memory for further processing. Alternatively, some transformations on the data were performed in functions that destroyed their variables once they were no longer in scope.

As part of our stop word filtering process, we employed stop word lists from three well-known libraries – Spacy, NLTK, and SciKit Learn – to filter out popular stop words. It was also decided that words would be filtered out by length and a good rule of thumb would be words that were three words or less. However, during the process of topic modeling, it was discovered that there were topic words that were three characters that held a lot of significance. For example, the word “God” in a thread titled “The health bill has PASSED!”

Using BERT with PySpark data frames also proved to be problematic when generating toxicity scores. This occurred even after creating User Defined Functions (UDFs) to process subject threads into toxicity scores. It is not clear why, but the process would frequently time out without generating any score values. It is believed there is an incompatibility with the way Toxic BERT and PySpark function. To work around this limitation, the PySpark data frame was first converted to a Pandas data frame for further processing.

Lastly, visualizing our findings proved tricky as the data in PySpark had to be duplicated to be brought into other libraries MatPlotLib and Wordcloud for visualizing. This placed further strain on the amount of available memory for further processing. In extracting the data from PySpark, we opted for very learn and simple data structures like lists to pass onto other libraries.

## Future Work Roadmap

The experience of the project left us with a lot of ideas on how we could expand the project. Some of those ideas include toxicity logic improvement, web application deployment, Gen-AI integration, and Integration with other social media platforms.

The toxicity logic we include only provides one number for overall toxicity. However, Toxic Bert can provide a series of toxicity categories indicating the type of toxicity observed in content, and to what degree. This could prove helpful for a moderator who wants to target a specific category of toxicity on the platform.

Now that we have a working method of analyzing the data, this could be wrapped in an easy-to-use interface for collecting a batch of data and running it through analysis in preparation for training a model. Developing a web application around this development pipeline would facilitate and speed up identifying toxic threads.

Integrating the output from this pipeline with a Generative AI can assist in incident report or threat report generation, assisting moderators with communicating findings. Moreover, it can help with notifying important stakeholders of the status of the situation.

Lastly, integration with other platforms is tantamount to not only providing as much coverage as possible but also ensuring that any model developed from the combination of topic modeling and toxicity score is robust.

# References

CNA. "Reddit may need to ramp up spending on content moderation, analysts say." 22 03 2024. *CNA.* 09 12 2024. <https://www.channelnewsasia.com/business/reddit-may-need-ramp-spending-content-moderation-analysts-say-4214801>.

Hanu, Laura. "Detoxify." 2022. *HuggingFace.* 09 12 2024. <https://huggingface.co/unitary/toxic-bert>.

HuggingFace. "Reeddit Question Best Answers." 2022. *HuggingFace.* 09 12 2024. <https://huggingface.co/datasets/nreimers/reddit\_question\_best\_answers>.

Reddit. "Reddit's Approach to Content Moderation." 12 09 2024. *Reddit.* 09 12 2024. <https://support.reddithelp.com/hc/en-us/articles/23511059871252-Content-Moderation-Enforcement-and-Appeals#:~:text=When%20our%20automated%20systems%20find,Policy%2C%20it%20is%20removed%20automatically>.

The Register. "Reddit rolling out AI bouncer to halt harassment." 07 03 2024. *The Register.* 09 12 2024. <https://www.theregister.com/2024/03/07/reddit\_introduces\_aipowered\_harassment\_filter/>.