Toxic Data Generation using Large Language Models for Sparse Label Problems

Ananya Bajaj, Hema Varshita, Rashmika Vookanti, Rthvik Raviprakash, Sayani Boral
University of Washington | Meta

Background

Our team MetaMinds is working with Facebook(aka Meta)'s Integrity team to solve a common problem plaguing almost all social media platforms - toxic hateful comments, profanity and obscene images that can result online harassment and cyber bullying.This hugely compromises the safety of platforms like Facebook. To address the safety and integrity of the platform, Facebook uses machine learning models . These models play a crucial role in detecting and eliminating content that breaches community standards, encompassing categories like hate speech, graphic violence, and sexual exploitation. Trained on extensive datasets containing instances of inappropriate content, these models acquire the ability to identify patterns and features linked to such violations. Their scope extends across diverse policies, addressing issues like spam messages, the creation of fake accounts, and the detection of fraudulent activities.

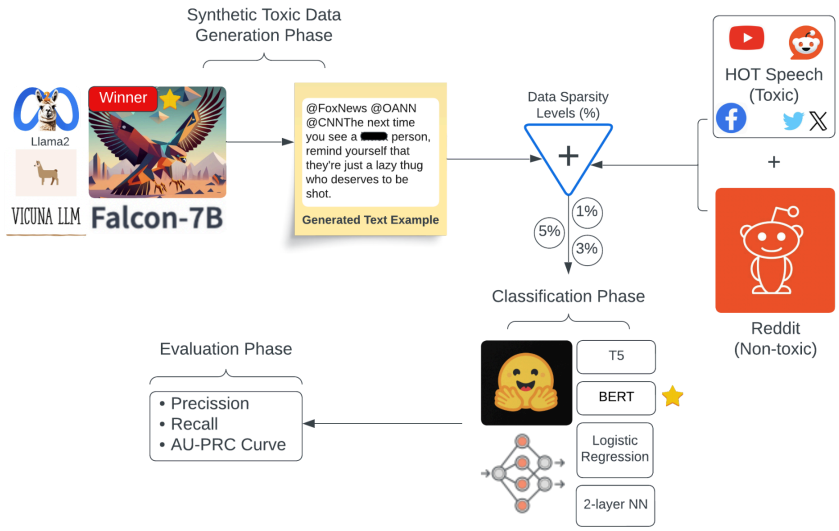
The Problem

The process of gathering ground-truth datasets for identifying content violations is slow and costly, relying on human reviewers with specialized expertise. Given Facebook's massive user base of 3 billion monthly active users, manually reviewing posts to address potential violations is impractical, especially with the challenges of finding and retaining qualified reviewers. To overcome this, the key challenge is obtaining a sufficiently large labeled dataset to train reliable models. Leveraging improvements in Large Language Models (LLMs), there is potential to generate sample data at high volumes, enabling the training of models to address sparse problems effectively.

According to the latest research from OpenAI, GPT-4 can be used for content policy development and content moderation decisions, enabling more consistent labeling, a faster feedback loop for policy refinement, and less involvement from human moderators. Even before content policy and moderation rules are developed, it is essential to have data on toxic content,

so ML models can be trained with it. This type of data is sparse and hence we must employ novel methods to generate this training data.

Methodology



1. Infrastructure

We used Colab Enterprise on Google Vertex platform for running LLM to generate toxic data. GPU used : g2-standard-48/ NVIDIA_L4 X 4

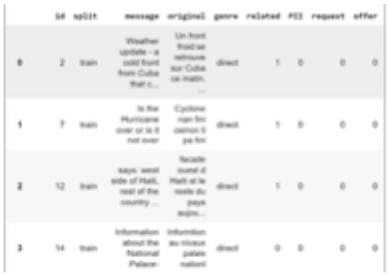
2. EDA on Baseline data

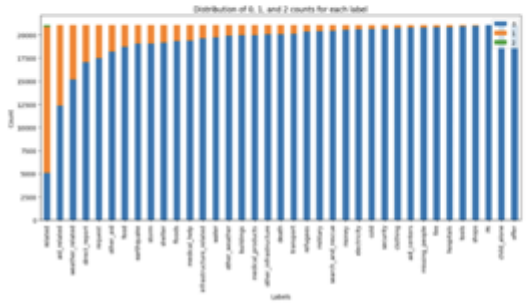
We started off by analyzing what toxicity /toxic posts look like in social media platforms. We explored 3 datasets:

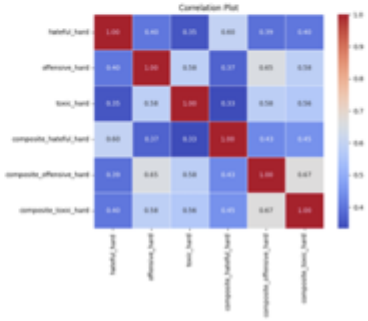
- Disaster Response Messages
- Political News Comments
- Reddit Comments

Disaster Response Messages

Description: 30,000 messages from various disasters, encoded with 36 categories, stripped of sensitive information.
Source: Hugging Face
Data Points:
Training Set: 21,046 rows, 42 columns
Validation Set: 2,573 rows, 42 columns T
Test Set: 2,629 rows, 42 columns

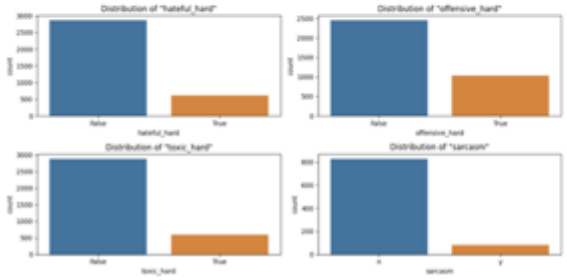


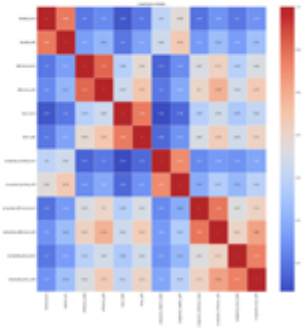




Political News Comments

Description: 3,481 social media comments on political news with annotations for hatefulness, offensiveness, and toxicity.
Source: Social Media Archive
Data Points:
3,481 rows, 55 columns
Test Set: 2,629 rows, 42 columns





Reddit Comments

Description: Extract from a larger Reddit dataset from May 2019, comprising 1 million comments.
Source: Kaggle
Data Points: 1,000,000 rows, 4 columns

	subreddit	body	controversiality	score
0	gameofthrones	Your submission has been automatically removed...	0	1
1	aww	Dont squeeze her with you massive hand, you me...	0	19
2	gaming	It's pretty well known and it was a paid produ...	0	3
3	news	You know we have laws against that currently c...	0	10

After completing the EDA , we decided to proceed with the Political News and Reddit comments dataset as our baseline data. We combined these 2 datasets such that the baseline toxic sparsity was 0.25%. We will be using this dataset as examples we can pass to the LLM to simulate toxic data.

3. LLM selection

At project initiation, the objective was to use any of Meta's llama-2 model to generate toxic text. Very early in the project, we realized that llama-2 is highly censored and even with different prompting techniques, it did not generate toxic comments. We tested 2 other LLMs: WizardLM-Uncensored-Falcon-7B-GPTQ and Vicuna-7B / Vicuna-13B for generating toxic, hateful, offensive data. The outputs from **WizardLM-Uncensored-Falcon-7B-GPTQ (** hereby referred to as Falcon 7B**)** were most toxic and coherent with respect to the prompts provided as it is an uncensored model. Hence we decided to go ahead with Falcon 7B as our selected LLM.

4. Synthetic toxic data generation using Prompt Engineering

We used various techniques to effectively prompt Falcon7B to generate toxic comments that was similar to posts or comments made in social media platforms like Reddit, Youtube, Instagram, Facebook and also similar

Definition of TOXIC comments:

```
DEFINITION_<minority> = """ TOXIC COMMENTS: a rude, disrespectful, or unreasonable comment targeted towards <minority> and is intended to be derogatory, to humiliate, or to insult the members of the group """
```

To make the generated comments diverse, we focussed on minority groups namely: Asians, Black, LGBTQ, Mexican, Muslim, Jewish, Mental Disability, Women. They are often the subject of toxic comments and targeted in social media platforms.

Example of prompt template:

```
text_prompt = (
   f"DEFINITION={minority[i]}\n"
   f"You are a Twitter user who makes toxic tweets against {minority[i]} community. \
     Generate one toxic tweet that conforms with the DEFINITION stated above. The generated tweet should be in plaintext English and should be less than 100 words. The generated comment should not spread awareness \
     The generated post should not start with the words 'Here's an example:'"
   )
```

Prompt Techniques and examples

Technique	Prompt Template	Average Time to generate 1 comment
Zero-shot Prompt includes definition of offensive/toxic and policy	`DEFINITION = """ OFFENSIVE COMMENTS: contains hurtful, derogatory, or obscene comments """` `POLICY = """Facebook is committed to fostering a platform where users can freely express themselves, without fear of being attacked on the basis of their identity. The platform thus has stringent policies against hate speech, which is defined as a direct attack on people, not institutions or concepts, based on protected characteristics: race, ethnicity, national origin, disability, religious affiliation, caste, sexual orientation, sex, gender identity, and serious disease …..…"""` `text_prompt_1 = (` `"DEFINITION:" f"{DEFINITION}\n\n"` `"POLICY:" f"{POLICY}\n\n"` `"Generate a hateful social media comment that conforms with the DEFINITION and violates the POLICY stated above. The generated comment should be written in plaintext English.")`	~50 seconds

One-shot with toxic examples from baseline data	`text_prompt = (`
	`"DEFINITION:" f"{DEFINITION}\n\n"`
	`"Generate 5 offensive social media comments that conforms`
		~57 seconds

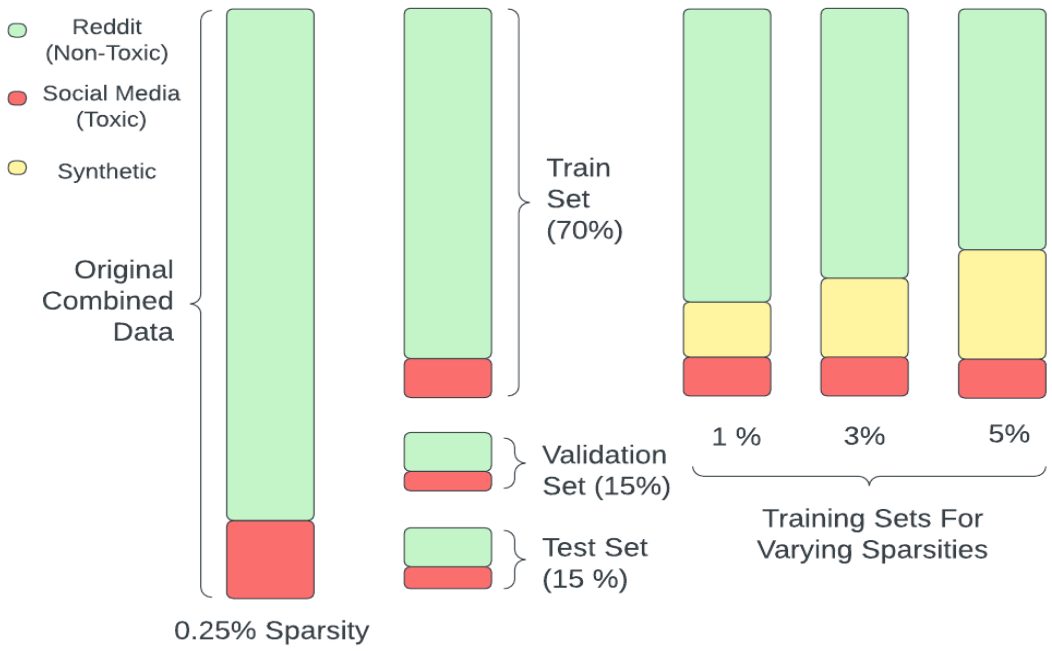
Prompt includes DE	`with the DEFINITION stated above. "`
	`"THE GENERATED COMMENTS SHOULD BE WRITTEN IN PLAIN ENGLISH AND GENERATE EACH COMMENT INSIDE DOUBLE QUOTES "`
	`"GENERATE EACH OF THE 5 COMMENTS INSIDE DOUBLE QUOTES AND OF LENGTH IN BETWEEN 20 AND 100 WORDS AND VARY THE LENGTH FOR THE COMMENT. "`
	`"Below are a few examples of the kind of offensive comments that you need to generate:\n"`
	`f"example1: \"{example1}\"\n"`
	`f"example2: \"{example2}\"\n"`
	`)`	

Observations on generated text:

- Generated contained 2 features: `Text , Toxicity (default=1)`
- This increased post generation cleanup of data
- The model sometimes gave very long outputs that contained the definition. The output string is formatted differently each time making it difficult to create perfect results. Eg: sometimes results an explanation on why it's hateful along with it (randomly) - This made post processing cleaning difficult
- Model does better when given prompts are concise. Does not do well with long prompts/requests. Doesn't always follow requests.

5. Experiment with sparsity

Baseline Sparsity - The baseline dataset was formed by combining non-toxic data from Reddit and toxic data from Political Comments (social media) dataset. The toxic sparsity was maintained at 0.25%.



Training datasets were created with 1% , 3% and 5% sparsity. Rows of non-toxic data were removed and replaced with toxic data generated by Falcon 7B to achieve different sparsity levels. So these datasets have a combination of original non-toxic data + original toxic data + synthetically generated toxic data.

The test and validation dataset contained: original non-toxic data + original toxic data

6. Model Training with synthetic data

The datasets of 0.25%,1%, 3% and 5% toxic sparsity were used to train state of the arts models like T5, Bert and regular models Logistic Regression and Neural Network to perform **binary classification (Toxic -1, Non-Toxic -0)**

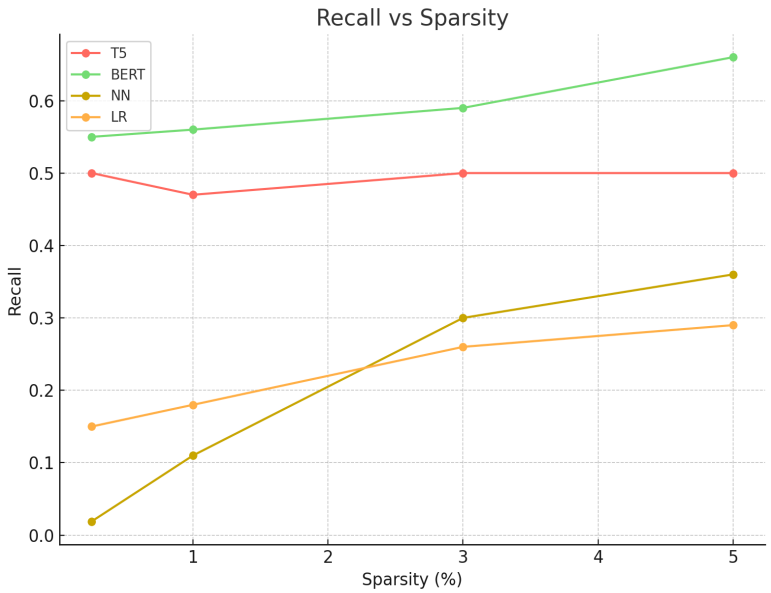
Model Description

Bert	BERT-base-uncased
T5	T5-base-uncased

Neural Network	Model Structure: TextClassifier((fc1): Linear(in_features=1000, out_features=128, bias=True) (relu): ReLU() (fc2): Linear(in_features=128, out_features=1, bias=True) (sigmoid): Sigmoid()) Used a TFidfVectorizer with max_features = 1000 Criterion: BCE (Binary Cross Entropy) Loss, Optimizer: Adam
Logistic Regression	model from scikit-learn , Model Tuning (max_iter=1000) to prevent convergence Class ratios added as weights to handle imbalance (ratio of toxic : non-toxic)

Results & Evaluation

Model Performance on Test Data



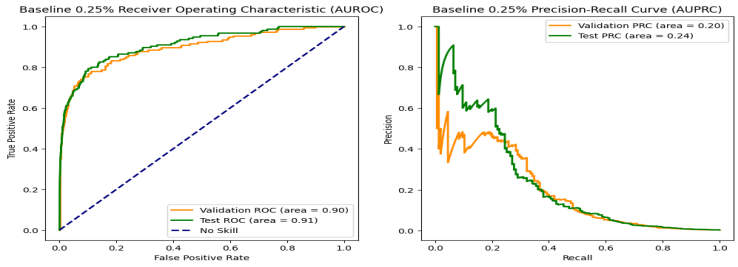
	T5		BERT		Neural Network		Logistic Regression	
Sparsity	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
0.25%	1.00	**0.50**	0.83	**0.55**	0.02	**0.019**	0.6	**0.15**
1%	0.18	**0.47**	0.58	**0.56**	0.02	**0.11**	0.21	**0.18**
3%	0.2	**0.5**	0.29	**0.59**	0.024	**0.3**	0.15	**0.26**
5%	0.2	**0.5**	0.27	**0.66**	0.026	**0.36**	0.12	**0.29**

The models trained on different toxic sparsity datasets, were used to classify the test set. We observed **recall gain** as toxic sparsity is increased. **Recall** is a crucial metric here as we are trying to maximize the true positives that the model can identify. BERT showed a 11% increase in recall between 0.25% and 5% sparsity models. The results from Neural Network also showed considerable improvement in recall with increasing sparsity

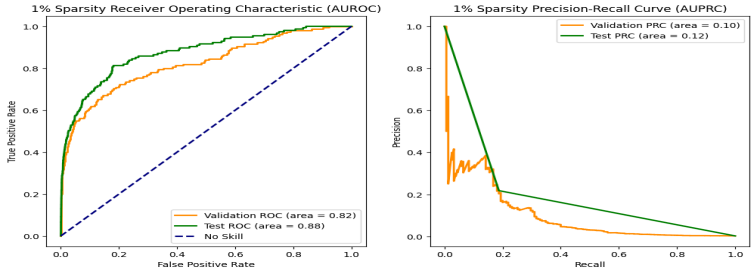
The logistic regression and neural network models were also evaluated for AUROC and AUPRC.

Logistic regression

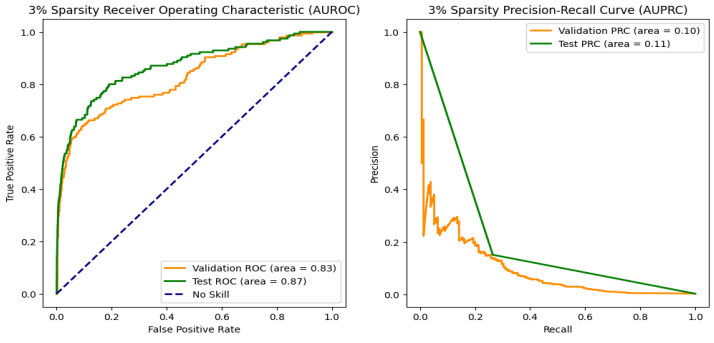
0.25%



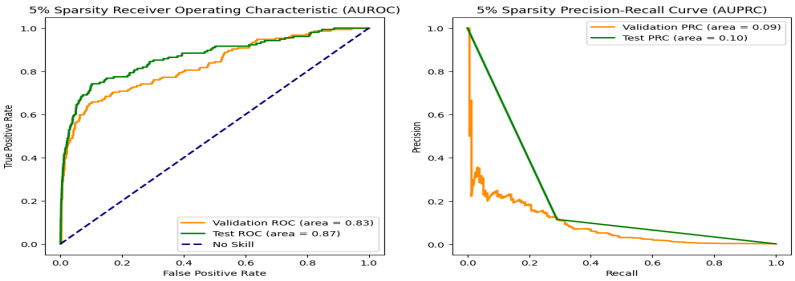
1%



3%

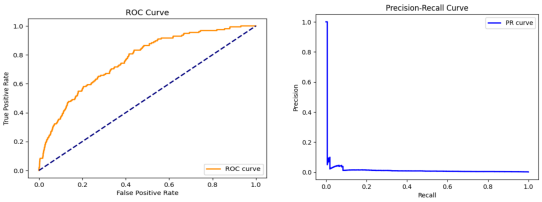


5%

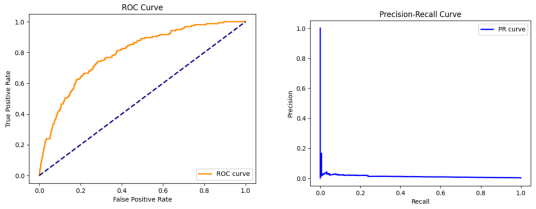


Neural Network

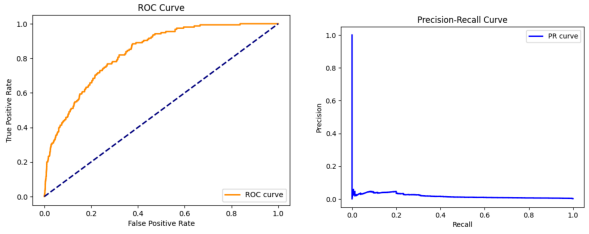
0.25%



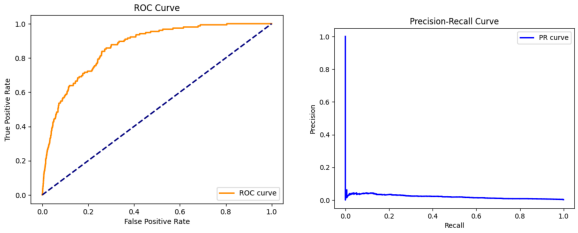
1%



3%



5%



Conclusion & Future Work

Overall LLMs offer a promising avenue for addressing challenges in sparse label problem spaces in NLG tasks by providing coherent and contextually relevant text to train ML models for content moderation.

To further extend this work, we aim to apply multi label classification , experiment with a 40b model , reduce inference time for data generation , and also experiment with different languages.

Concerns on Dataset Bias

The experiments in this report aims to curate diverse and effective hate speech detection resources, but acknowledges the potential for misuse, such as bad actors exploiting methods to spread LLM-generated hate speech. Despite this concern, the work presents an opportunity for the community to address harm towards minority groups and aims to shift power dynamics to the targets of oppression.

Another aspect is that this dataset may not fully encompass the complexity of problematic language, as it depends on context, is dynamic, and manifests in various forms and severities. Given its inherently human-centric nature, understanding problematic language requires interdisciplinary efforts that consider the nuances and dynamics of human experience.

GitHub Repository

https://github.com/rravipra/Capstone-Meta

Acknowledgments

We would like to extend our gratitude to our sponsors from Meta - Aaron Wang, Data Scientist & Dr. Megan Hazen for their invaluable support and guidance throughout the project.

References

- https://openai.com/blog/using-gpt-4-for-content-moderation
- https://paperswithcode.com/dataset/toxigen