Computer Science 1090B: Final Project Milestone 3

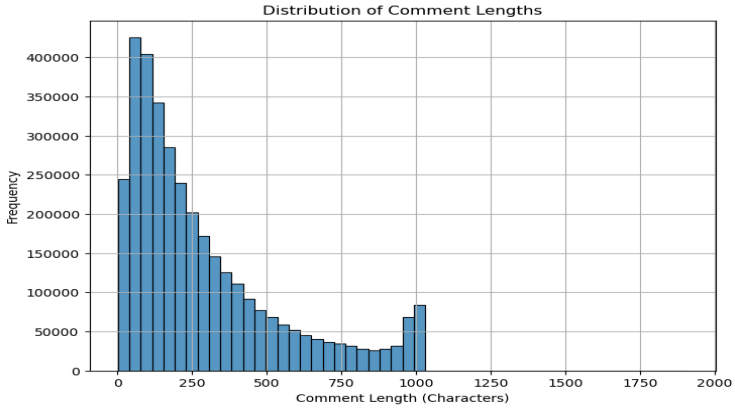
Amar Boparai, Andrew Lobo, Conrad Kaminski, Xiaoxuan Zhang, Xuanthe Nguyen
Canvas Group 31

I. Data Description & Addressing Feedback

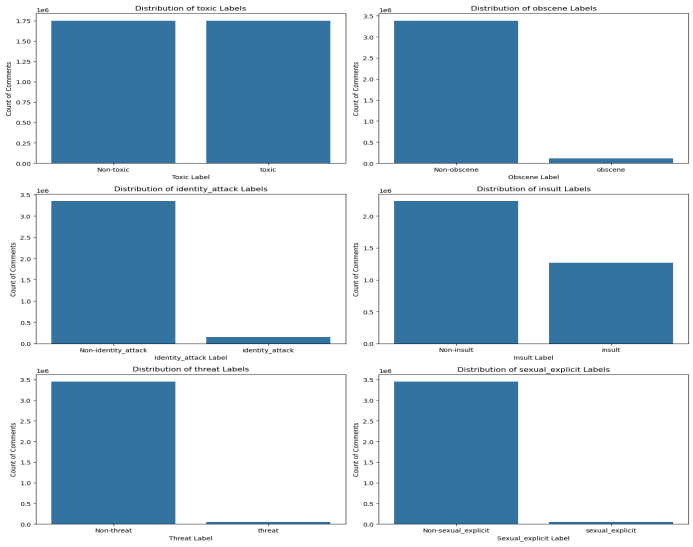
Our feedback from MS2 asked how we plan to evaluate the fairness of our model using the identity columns (e.g., `female`, `black`, `muslim`, etc.). These columns indicate whether a comment references that specific identity. These identities will not be used during training, but we will use it in our post-training fairness evaluation. We discuss more in our .ipynb file.

Once again, we are using the Jigsaw Unintended Bias in Toxicity Classification dataset from Kaggle, which contains ~1.9 million online comments labeled for various forms of toxicity. In MS2, after cleaning and subsetting, our final dataset includes comment_text as input and seven binary toxicity labels as outputs (toxic, insult, threat).

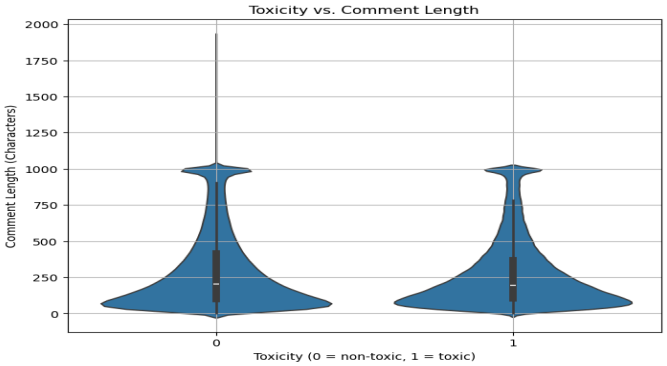
II. Exploratory Data Analysis



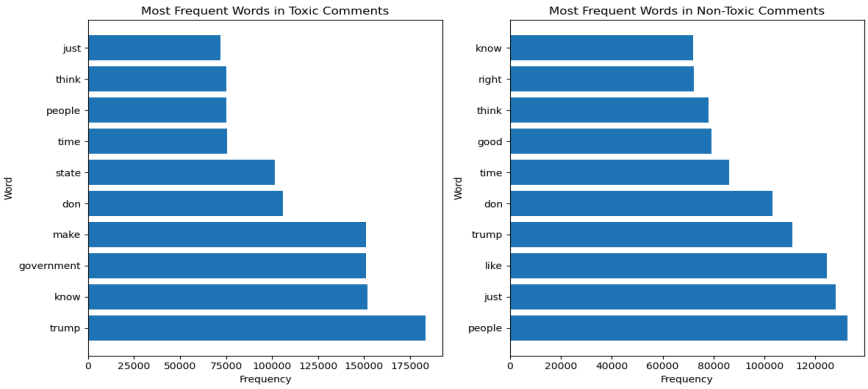
The histogram shows a right-skewed distribution. Most comments are relatively short (under 250 characters). After 1000 characters, there is almost a complete drop off.



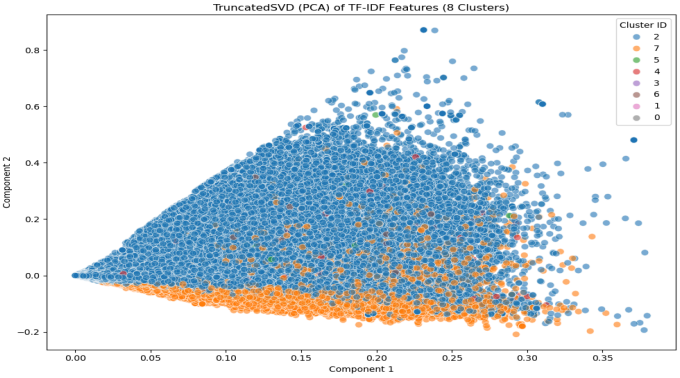
The labels, with the exception of `toxic`, are heavily imbalanced (after balancing `toxic` class with over-sampling — though overbalanced). `insult` also seems to be relatively balanced. We will not be balancing the other predictors other toxicity labels because our predictive model is solely focused on predicting the toxic label, and the other toxicity labels are being used mainly for evaluation purposes. We will discuss how we plan on addressing the other class imbalances in our baseline model implementation plan.



The violin plot shows that the comment length for both toxic and non-toxic comments are fairly similar. The median comment length in both seems to be around 250 characters. We can see, then, that toxicity does not seem to be very dependent on comment length.



Words like "trump," "don," "time," "people," "just," and "think" appear in both toxic and non-toxic comments. We wonder if "trump" refers to President Trump and "don" refers to Donald. What is interesting is that certain words like "government" and "state" appear frequently in toxic comments. Political topics may include a lot more toxicity in this dataset. Then, words like "like," "good," and "right" appear more often in non-toxic comments.



The above PCA is the result of an idea from ChatGPT on how else we could visualize our data. By separating into our known number of clusters (8 total including the majority non-toxic), we can see that non-toxic dominates, but there's an interesting trend with category 7 as well. The rest are all scattered, which makes sense, since all of these toxic posts will likely have similar key words or characteristics no matter if they are insulting, obscene, etc. It would require a lot more semantic work to distill these clusters down to separate issues, especially given their sparsity in the data.

III. Baseline Model Implementation Plan

Because the only feature we could use is the user comment, we can treat this problem as a NLP classification problem. We plan to use a version of BERT (e.g. DistilBERT) as the baseline for classification. The BERT model's attention mechanism allows it to understand the entire comment, and is the most commonly used model for NLP classification in the industry. In terms of Train/Test data, we noticed that the dataset is highly imbalanced. Hence, we currently plan to oversample positive samples. We could use LLM to rewrite the positive samples, so we increase the variety of our positive sample dataset. Eventually, we aim to have a positive:negative ratio of 1:3. In terms of evaluation, because the dataset is imbalanced, metrics such as binary cross entropy loss are less effective at measuring the effectiveness of the model. We should focus more on the following metrics:

- Normalized entropy = CrossEntropy(Model) / CrossEntropy(Baseline)
    - The baseline model is a naive model that predicts p as the average probability over the training dataset. This loss deals with class imbalance better. For example, if there is 1% positive data and 99% negative data, CrossEntropy(Baseline) would already be pretty low. Normalized Entropy would account for this issue.
- Precision (TP/(TP + FP))
- Recall (TP/(TP + FN))
- F1 score
- Area Under the Precision-Recall Curve (AUPRC)

How do we decide the correct threshold to classify whether a comment is toxic? We think that the choice depends on how we plan to use the classifier and how much we care about false positives and false negatives. For example, in a production setting, having false positives may result in too many comments being deleted incorrectly, which discourages users from making comments. Having false negatives would result in toxic comments not being flagged/deleted. There are two methods we can try to decide the threshold:

- Solution 1: Set a minimum recall target (e.g., "We must catch at least 99.5% of all toxic comments") Find the highest threshold that still meets our minimum recall target.
- Solution 2: Cost-sensitive thresholding. False positives and false negatives have different costs (e.g. false positive=1, false negative=2). Choose a threshold that minimizes the cost.

Lastly, because we oversampled positive labels, our model's prediction on actual traffic/test dataset would not be well-calibrated: it would predict a positive label more frequently than we expect. We could calculate a weight to be multiplied with model prediction to account for the fact that the training dataset's label distribution is different from production. An example formula is:

$$p_{\text{true}} = \frac{\alpha \cdot p_{\text{pred}}}{(1-p_{\text{pred}})+\alpha \cdot p_{\text{pred}}}.$$

Question: How can we predict the toxicity of online comments, identify the factors contributing to toxicity, and evaluate the fairness of our model across different identity attributes?