Domain Background

The internet can feel like a safe platform for communication. It strips the need for a face-to-face conversation. It enables us to be identified from a random nickname or profile picture and keeps our true identities hidden from others. However, it is these characteristics of the internet that has made it a far from safe place to hold public conversations.

The increasing degree of conversational toxicity and online bullying by ‘keyboard warriors’ is concerning. The threat of online abuse and harassment can hinder people from genuinely expressing themselves and seeking opinions from others. In more severe cases, it can negatively impact the mental and psychological well-being of those being bullied online.

Problem Statement

It is therefore of value to be able to effectively identify toxicity in online postings, especially user comments. It enables platforms to facilitate healthier and safer online discussions for their users, which is a win-win situation for both users and the platforms. In this project, we will be building multi-headed machine learning models to detect toxicity in online comments. There are 6 class labels for classification of comments, namely, toxic, severe toxic, obscene, threat, insult and identity-hate.

Dataset and Input

This project leverages the dataset used in Kaggle’s Toxic Comment Classification Challenge[1], provided by the Conversation AI team, a research initiative founded by Jigsaw and Google. The dataset consists of comments from Wikipedia’s talk page edits. There are 159,571 training examples and 153,164 testing examples. Out of the testing examples, only ~42% of them have labels and would be used for scoring and evaluation. Refer to Table 1. for data dictionary.

Table 1. Data dictionary of dataset

Data	Description	Schema
train.csv (159,571 rows x 8 fields)	Wiki talk page edit comments in the train dataset Note that examples where all class labels take value 0 are considered non-toxic	- id: str - comment_text: str class labels - toxic: int (0 or 1) - severe_toxic: int (0 or 1) - obscene: int (0 or 1) - threat: int (0 or 1) - insult: int (0 or 1) - identity_hate: int (0 or 1)
test.csv (153,164 rows x 2 fields)	Wiki talk page edit comments in the test dataset	- id: str - comment_text: str
test_labels.csv (153,164 rows x 7 fields)	Labels for test dataset Note that if all class labels are -1, it is not used for scoring and evaluation	- id: str - toxic: int (0 or 1 or -1) - severe_toxic: int (0 or 1 or -1) - obscene: int (0 or 1 or -1) - threat: int (0 or 1 or -1) - insult: int (0 or 1 or -1) - identity_hate: int (0 or 1 or -1)

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

Solution Statement

Classification models would be built for this multi-class classification problem. Specifically, the strategy would be to build different models and make predictions separately for each class of toxic comments. This would be equivalent to performing a binary classification for each class.

Benchmark Model

Since we are essentially performing a binary classification for each class, logistic regression model would serve as a good and simple model to start with for this problem. Coupled with a bag-of-words/features representation of the text comments, it would serve as the baseline model for this problem.

Evaluation metrics

Mean column-wise ROC AUC would be used as the evaluation metric, adapted from the evaluation criteria of the Kaggle challenge itself. This would be computed by the average of AUC scores of the 6 class labels.

Project Design

An exploratory data analysis (EDA) would first be performed on the dataset. Specifically, missing values checks would be performed. It is important to either discard these data from training or impute them with a fixed value so that it would not affect model training. Class distribution of the training dataset would also be explored. There is likely going to be a severe class imbalance, with non-toxic comments forming the majority. Correlation is another common check performed during EDA; Crammer's V statistic would be used to look at potential correlations between class labels. It would also be interesting to note most commonly occurring words in the various class labels.

Moving on to modelling, following benchmark modelling, another model of consideration for this problem is the NB-SVM proposed by in a 2012 paper by Wang and Manning[2] that scales word count features with Naïve Bayes probabilities in a linear model such as SVM (or logistic regression). Different text-preprocessing techniques such as stopwords removal and lemmatization would be used and compared. Modelling would also be explored using TF-IDF representation of the data and compared to the original bag-of-words/features representation.

Given that deep learning has gained popularity over the years in effectively solving many different types of machine learning problems, it would be of interest to explore how neural network models perform on this dataset. Simple neural network models based on LSTM and GRU architecture would be built and compared against each other. Initialization of the neural network models with pre-trained word embeddings such as Glove embeddings would also be explored.

Finally, the modelling results would be compared, and the project would be concluded.

[2] SidaWang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012), pages 90–94, Jeju Island, KR, 2012.