TOXIC COMMENTS CLASSIFICATION

Presented By:

DHANUSH BALAJI G
III Year, KVCET
NM ID: au421221104012
email id: dhanushbalaji1204@gmail.com

PROBLEM STATEMENT

Toxic comment classification is a challenging task in natural language processing (NLP) that aims to automatically identify and categorize toxic or offensive comments in online conversations. These comments can range from hate speech and harassment to threats and profanity, and they can have serious consequences such as spreading negativity, inciting violence, or causing harm to individuals or communities. Detecting and filtering out toxic comments is crucial for maintaining a healthy and respectful online environment.

PROBLEM GOAL

The goal of this project is to develop a machine learning model capable of accurately classifying comments as toxic or non-toxic. Specifically, the model should be able to identify various types of toxicity, including but not limited to, threats, insults, obscenities, and hate speech.

APPROACH

1. Data Preprocessing: Clean and preprocess the text data by removing noise, punctuation, special characters, and standardizing text (lowercasing, stemming, lemmatization).
2. Feature Engineering: Extract relevant features from the text data, such as word embeddings (e.g., Word2Vec, GloVe), TF-IDF vectors, and n-grams.
3. Model Selection: Experiment with various machine learning algorithms (e.g., logistic regression, random forest, support vector machines) and deep learning architectures (e.g., recurrent neural networks, convolutional neural networks) to identify the most suitable model for the task.

4. Model Training: Train the selected model on the preprocessed text data, tuning hyperparameters using techniques such as grid search or random search.
5. Model Evaluation: Evaluate the trained model on a separate validation set using appropriate evaluation metrics.
6. Fine-tuning: Fine-tune the model using techniques such as transfer learning or ensemble methods to further improve performance.
Model Deployment: Deploy the trained model to classify new comments in real-time or batch processing environments, ensuring scalability and efficiency.

CHALLENGES

- Dealing with imbalanced datasets where the number of toxic comments is much smaller than non-toxic ones.
- Handling diverse types of toxicity and ensuring the model's ability to generalize across different forms of toxic behavior.
- Addressing the dynamic nature of language and emerging toxic patterns, requiring continuous model monitoring and updating.

FEATURE WORK

Text Embeddings: Experiment with different word embedding techniques such as Word2Vec, GloVe, FastText, and BERT embeddings to capture semantic meaning and contextual information of words more effectively. Pre-trained embeddings can be fine-tuned during model training to better suit the specific task of toxic comment classification.

Character-Level Features: Besides word-level features, consider incorporating character-level features such as character n-grams or character-level embeddings. This can help the model capture morphological and syntactical patterns, especially in handling misspelled or slang words common in online comments.

Topic Modeling: Explore techniques like Latent Dirichlet Allocation (LDA) or Non-negative Matrix Factorization (NMF) to extract topics from the comments. By identifying underlying topics or themes, the model can better understand the context of the comments and improve classification accuracy.

MODELLING

Data Splitting: Begin by splitting your dataset into three subsets: training, validation, and test sets. The training set is used to train the model, the validation set is used for hyperparameter tuning and model selection, and the test set is used to evaluate the final model's performance.

Advanced Models:Experiment with more advanced machine learning algorithms such as Random Forest, Gradient Boosting Machines (GBM), or XGBoost. These models can capture non-linear relationships and interactions in the data more effectively.

Deep Learning Models:

- Implement deep learning architectures such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or Transformer-based models like BERT.
- Preprocess text data and tokenize it appropriately for input into deep learning models.
- Experiment with different architectures, layer configurations, and regularization techniques to find the most effective model.
- Train the deep learning models on the training data and monitor their performance on the validation set.

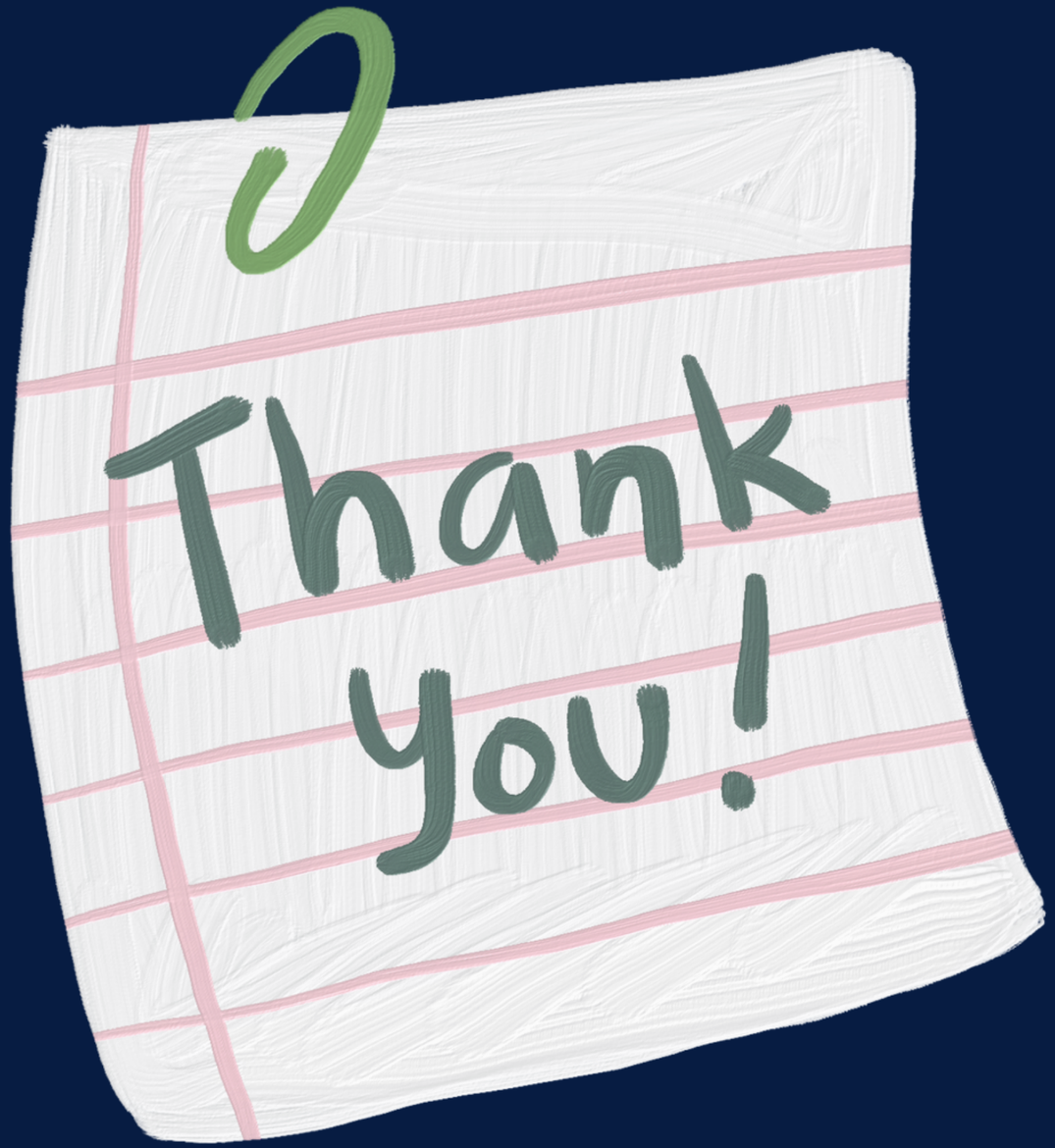
Model Deployment:

- Once satisfied with the model's performance, deploy it in production for real-time or batch processing of new comments.
- Implement appropriate monitoring and logging mechanisms to track model performance and detect drift or degradation over time.
- Continuously update and improve the model based on user feedback, new labeled data, or advances in modeling techniques.

RESULTS

- **Accuracy:** Accuracy measures the proportion of correctly classified comments out of all comments in the dataset. While accuracy provides an overall view of model performance, it might not be sufficient for imbalanced datasets where the classes are unevenly distributed.

- **Confusion Matrix:** A confusion matrix provides a detailed breakdown of the model's predictions, showing the number of true positives, true negatives, false positives, and false negatives. It helps in understanding where the model succeeds and where it fails in classification.


Thank
you!