TITLE -COMMENT ANALYZER

Sentiment Analysis and Toxicity Detection in Comments using Machine Learning

ABSTRACT

This aims to develop a robust analytical system for comments, primarily focused on sentiment classification (positive, negative, neutral) and toxicity detection (hate speech, insults, threats). For this purpose, we will use NLP techniques and machine learning algorithms in order to process and analyze the text data; extract meaningful features from these texts; and finally, develop predictive models. Due to the practical applications, the system may be used for monitoring social media, providing customer service, and managing a community that uses the Web for their interface.

This has broad applications across various domains, including social media monitoring, customer feedback analysis, market research, and literature analysis

DATASET

We will use a mix of available datasets and gather our own annotated data. Some of the potential datasets include:

Kaggle Toxic Comment Classification Challenge: This is a huge dataset containing a massive collection of comments from Wikipedia talk pages, labeled for toxicity.

GoEmotions: A dataset of YouTube comments annotated with fine-grained emotion labels.

Sentiment140: A dataset of 1.6 million tweets labeled with positive, negative, or neutral sentiment.

We might also gather and annotate a much smaller, domain-specific dataset specifically related to an application of interest.

ALGORITHM

We will explore and compare the performance of a number of machine learning algorithms, including:

Support Vector Machines (SVM): A powerful classifier, known for its efficiency and accuracy.

Naive Bayes: A probabilistic algorithm which is very simple and efficient in text classification tasks.

Random Forest: An ensemble learning technique that combines multiple decision trees to improve accuracy and robustness.

Recurrent Neural Networks (RNNs), especially LSTMs: Deep learning models that are particularly suitable for sequential data like text, capturing long-term dependencies.

We will preprocess the text data using techniques

such as: Tokenization: Splitting text into individual

words or sub-words.

Stop word removal: Removing common words that do not carry significant meaning (e.g., "the," "a," "is").

Stemming/Lemmatization: Reducing words to their root form (e.g., "running" -> "run").

TF-IDF: A scheme of feature weight for term that is very selective of the less frequent words within a document which are essential. Word Embedding (Word2Vec, GloVe): They are a word representation using vector and represent rich semantic relations in between words.

TEAM MEMBERS

S.Samanvitha-2320030437
M.Annapurna-2320090064