SVKM's NMIMS University

(Deemed-to-be University)

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING (MPSTME)

An Ensemble Model for Contextual Aggression Detection in Hinglish

Siddhi Kadam (A156)
Asika Babydasan (A150)
Vaishnavi Awasthi (A126)

Faculty Mentor:
Madhura Vyawhare

1. Introduction & Problem Statement

Online aggression is a significant threat to digital communities. While simple systems can filter overt aggression (e.g., slurs), they often fail to detect **covert aggression**, which includes sarcasm, passive-aggression, and micro-aggressions. This nuanced, context-dependent hostility evades basic keyword flagging.

This challenge is severely amplified in **Hinglish**, a code-mixed language blending Hindi and English that is prevalent on social media. Traditional NLP pipelines, built for monolingual text, often break when processing this informal mix.

The goal of this project is to implement and evaluate a model for a three-class classification task that requires true contextual understanding:

- **NAG (Non-Aggressive):** Neutral or positive conversation.
- **CAG (Covertly Aggressive):** Sarcasm and passive-aggression. This is the primary challenge.
- **OAG (Overtly Aggressive):** Obvious insults, slurs, and direct threats.

2. Dataset

This project utilized the **TRAC 2018 (Trolling, Aggression and Cyberbullying) dataset**. This dataset was ideal as it consists of real-world social media posts and comments in Hinglish and is already labelled for the NAG, CAG, and OAG classification task.

The dataset contains a total of **15,000 entries**, which were split as follows:

- **Training Set (12,000 entries):** This set was further subdivided, with 80% (9,600 entries) used for training the base models and 20% (2,400 entries) used for validation and training the stacker.

- **Test Set (3,000 entries):** This set was held out for the final evaluation of the ensemble model.

3. NLP Methodology

The core of this project is a **stacking ensemble** that combines the strengths of diverse NLP feature sets. No single feature captures the full problem, so multiple "base learners" are used in parallel.

3.1 Text Preprocessing Pipeline

Raw social media text is messy and requires a robust cleaning pipeline. Based on the provided notebook and presentation, the steps include:

- **Standard Cleaning:** Text is lowercased, and all URLs and user mentions (@username) are removed.
- **Emoji Normalization:** Emojis are converted to their text representations (e.g., 😂 becomes `:face_with_tears_of_joy:`). This preserves the emoji's meaning as a feature.
- **Hinglish Handling:** Joined hashtags are split into constituent words (e.g., `#deshbhakt` -> "desh bhakt") using the `wordsegment` library.
- **Punctuation Handling:** Most punctuation is removed, but `!` and `?` are strategically kept, as their presence is a strong indicator of sentiment or aggression.
- **Abuse Tokenization:** A key step is the normalization of common Hinglish and English expletives (e.g., `ch*t*ya`, `mc`, `bc`) to a single, unified **ABUSE** token. This helps the model recognize overt aggression without needing to learn every single variant.

3.2 Feature Engineering

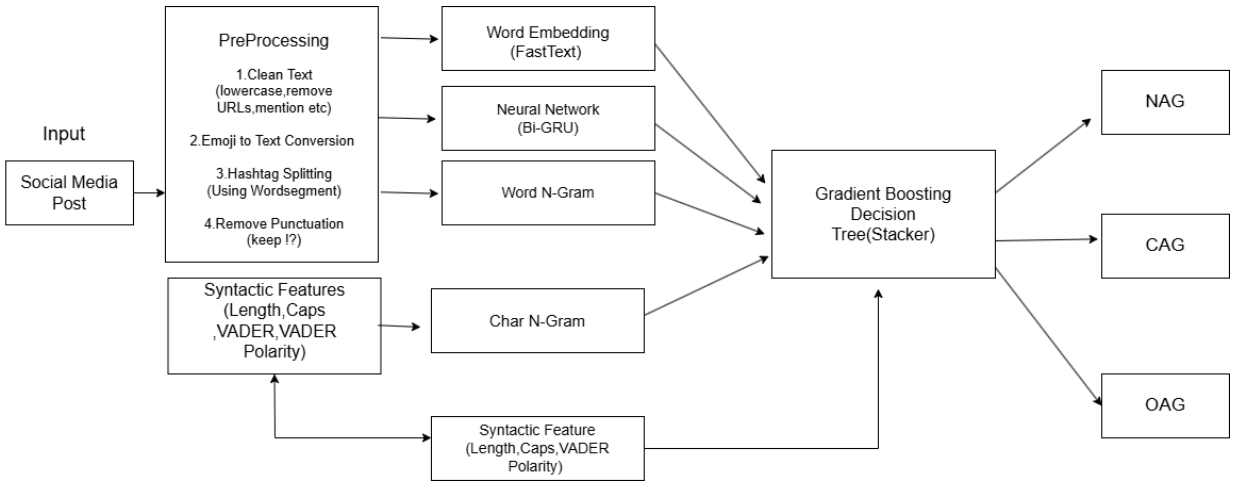
Four distinct sets of features were engineered, each trained with its own base model.

1. **Lexical Features (Word N-Grams):**
    - **Technique:** A TF-IDF Vectorizer was used to analyze the frequency of word 1-grams and 2-grams (word pairs).
    - **Purpose:** This model is highly effective at catching common, overt phrases and insults (e.g., "go away").
    - **Classifier:** A Logistic Regression model was trained on these TF-IDF features.
2. **Sub-word Features (Character N-Grams):**
    - **Technique:** A TF-IDF Vectorizer was used on sequences of characters (ranging from 2 to 6 characters long).
    - **Purpose:** This model is extremely robust against **misspellings, abbreviations, and evasion tactics** (e.g., "a$$hole" or "stup*d"), as it finds patterns *inside* words.
    - **Classifier:** A Logistic Regression model was also used here.
3. **Syntactic & Sentiment Features:**
    - **Technique:** This model ignores the words themselves and focuses only on the *style* of the text. The features include:
        - **VADER Sentiment:** Polarity scores (positive, negative, neutral, compound).
        - **Syntactic:** Text length, proportion of UPPERCASE letters, and counts of `!` and `?`.
    - **Classifier:** A Logistic Regression model was trained on this metadata.
4. **Semantic Features (Deep Learning):**
    - **Technique:** A Bidirectional GRU (Bi-GRU) neural network with FastText word embeddings was used as the fourth base learner.

- **Purpose:** Its role is to capture the complex **semantic meaning, context, and word order** that N-gram models miss, which is crucial for identifying nuanced covert aggression.

3.3 Stacking Ensemble Architecture

The final model is a **Gradient Boosting Decision Tree** (the "Stacker"), as shown in the provided architecture diagram .



1. A social media post is fed into the system.
2. It's processed in parallel by all the feature extractors (Word N-Gram, Char N-Gram, Syntactic, Bi-GRU).
3. The predictions from these four base models, along with the raw syntactic features, are collected.
4. These outputs are used as the *new input features* for the Gradient Boosting "Stacker" model.
5. This meta-model (Stacker) learns from the strengths and weaknesses of its base learners to make a final, more accurate classification.

4. Results

The final ensemble model was evaluated on the held-out test set. The model achieved a **weighted F1-score of 0.61**.

The detailed per-class performance is as follows:

```
============================================================
--- FINAL ENSEMBLE EVALUATION ON TEST SET ---
============================================================
[LightGBM] [Warning] feature_fraction is set=0.45, colsample_b
[LightGBM] [Warning] bagging_fraction is set=0.8, subsample=1.
              precision    recall  f1-score   support

     NAG (0)       0.55      0.66      0.60       538
     CAG (1)       0.58      0.59      0.59      1246
     OAG (2)       0.66      0.60      0.63      1217

    accuracy                           0.61      3001
   macro avg       0.60      0.62      0.61      3001
weighted avg       0.61      0.61      0.61      3001
```

As hypothesized, the model performed best on Overt Aggression (OAG) with an F1-score of 0.63. The most challenging class was **Covert Aggression (CAG)**, which achieved an F1-score of 0.59. This highlights the difficulty of detecting nuanced sarcasm and subtext compared to overt insults.

5. Conclusion

This project demonstrates that contextual aggression in Hinglish is a solvable, though difficult, NLP problem.

The proposed ensemble architecture, which **combines traditional NLP features (N-Grams, Character-Grams, VADER)** with a semantic deep learning model, proves to be an effective

baseline. The success of the Gradient Boosting stacker shows the value of integrating diverse feature sets-lexical, sub-word, syntactic, and semantic-to build a robust classifier for a complex task.