Bot Detection System - Technical Documentation

Project Overview:

This system is designed to detect bots on Twitter by combining text features derived from tweet content
and behavioral features gathered from the user activity. The model uses a hybrid architecture combining
BERT-based text embeddings with manually engineered behavioral features.

Components:

- TextProcessor: Uses BERT to extract text features from tweets.
- BehavioralFeatureExtractor: Extracts behavioral features from Twitter data, including post frequency,
  engagement, location information, and more.
- BotDetectionModel: A neural network that combines text and behavioral features for bot classification.
- BotDetector: Main class responsible for training, evaluation, and inference.

Features:

- BERT-based feature extraction from tweets for contextual understanding.
- Various engagement metrics like retweets, mentions, and follower count.
- Custom behavioral features based on user activity patterns.
- Model training with early stopping and validation metrics.

Tech Stack:

- Python (3.x)
- PyTorch for deep learning model implementation
- Transformers library by Hugging Face for BERT-based models
- Pandas, NumPy for data manipulation and preprocessing
- scikit-learn for feature scaling and evaluation metrics
- FPDF for generating PDF documentation

Data Flow:

1. Data Preprocessing: The raw data from Twitter (CSV) is processed to extract relevant features.
2. Feature Extraction:
   - Textual features are extracted using a pretrained BERT model.
   - Behavioral features are computed based on user engagement and activity data.
3. Model Training: The model is trained with both textual and behavioral features.
4. Evaluation: The model is evaluated using various metrics such as accuracy, precision, recall, and F1 score.
5. Inference: The trained model is used to classify whether a user is a bot based on a given tweet and behavioral data.

Model Details:

- The model uses a hybrid approach: combining textual data (via BERT) and behavioral data (via custom features).
- The neural network consists of two branches: one for text features and another for behavioral features.
- Both branches' outputs are concatenated and passed through further layers for classification.
- The model outputs a probability (between 0 and 1) indicating the likelihood that the user is a bot.

Training:

- Loss function: Binary Cross-Entropy (BCELoss)

- Optimizer: AdamW

- Training includes mixed-precision training using AMP for faster computation.

Licensing:

Contact:

For further inquiries, please contact Utkarsh Verma.