Bot Detection on TwiBot-22

Project overview

Natural Language Processing

Massimiliano Capparuccia (65918A)

December 23, 2025

UNIVERSITÀ
DEGLI STUDI
DI MILANO

Table of Contents

Context & Goals

- **Context:** The proliferation of fake profiles on social media poses significant threats to information integrity.
- **Project Goal:**
  - Investigate if a **lightweight model** can detect these accounts using linguistic features, combined with structured metadata.
  - The approach prioritizes **efficiency**: the entire pipeline was executed on consumer hardware without external GPU resources.

Research Question

1 Introduction

Core Question

To what extent can **language cues** and simple **multimodal footprint features** discriminate bots from humans in a contemporary Twitter/X dataset?

Our Approach:

- **Objective:** Binary classification of user profiles (Bot vs. Human).
- **Model:** A hybrid system combining semantic text embeddings and structured account signals.

Dataset

2 Overview

*”**TwiBot-22**, a comprehensive graph-based Twitter bot detection benchmark that presents the largest dataset to date, provides diversified entities and relations on the Twitter network, and has considerably better annotation quality than existing datasets.”*

- **Split Protocol:** I used the official dataset split provided in the repository to ensure reproducibility and fair comparison.
- **Why Twibot-22:** It is currently the largest and most comprehensive benchmark available, that allow for rigorous performance comparison.
- **Official Split:**
  - **Train:** 700,000 users
  - **Validation:** 200,000 users
  - **Test:** 100,000 users

Text Encoding

Text Encoder

I use the **sentence-transformer** model: `paraphrase-multilingual-MiniLM-L12-v2` (384d).

- **Why this model:** It is a state-of-the-art Transformer architecture that offers high performance with low computational overhead and high inference speed
- **Bio Embedding:** A 384-d vector representation of the user profile description.
- **Strategy:** I average the embeddings of the user's last N tweets, with $N = 20$ (Mean Pooling).

Model Architecture

2 Overview

Classifier:

- **Algorithm:** Gradient-Boosted Decision Trees (XGBoost).
- **Features:** Concatenation of Structured Metadata + Bio Embedding + Tweet Embedding.

Training Protocol:

- Trained on the TwiBot-22 train split.
- Used early stopping rounds to prevent overfitting
- **Thresholding:** Optimized decision threshold (0.1557) based on validation set F1-score.

Table of Contents

3 Results

Performance Overview
3 Results

Our full multimodal model achieved significant results on the test set:

Key Metrics

- **Accuracy:** 70.3%
- **F1 (Bot):** 57.8%
- **AUC-ROC:** 0.770

Official Baselines

Model	Acc.	F1
Random Forest	0.764	0.587
BotRGCN	0.797	0.575
RoBERTa	0.721	0.205
BERT GAT	0.719	0.211

Comparison:

- Outperforms several TwiBot-22 feature-based baselines in F1-score.
- Beats the language-only baseline (ROBERTa) significantly in F1 (57.8% vs 20.5%), demonstrating better bot sensitivity despite slightly lower accuracy.

Ablation Study

To understand the contribution of language features, i tested three variants:

Model Variant	Accuracy	F1 (Bot)
(S) Structural Only	60.1%	53.8%
(B) Bio + Structural	63.8%	55.0%
(F) Full (Tweets + Bio)	**70.3%**	**57.8%**

Key Finding: Adding aggregated tweet content boosts F1 by ≈ 4 points. This highlights the discriminative value of linguistic features the embeddings capture crucial behavioral patterns that metadata misses.

Feature Importance

- **High Gain:** Metadata like `followers_count` provides 3x more gain than any other feature.
- **Low Gain, High Impact:** Text embeddings rank low in ”gain” but are essential for the F1-score.

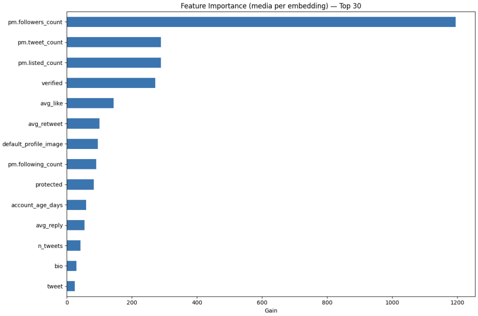

Table of Contents

4 Conclusions

Summary & Limitations

Summary:

- A lightweight approach achieves good results in identifying automated footprints.
- Linguistic signals are fondamental for improving detection beyond simple metadata.

Limitations:

- **Precision:** The model achieves a Precision of 0.50, indicating a high rate of False Positives.

Future Work

4 Conclusions

- **Graph Integration:** Incorporate follower/friend network structures to analize community structures and detect coordinated bot activities that are invisible when looking at users in isolation.
- **Image Integration:** Download all profile images in the dataset and use **Vision Transformers** (e.g., CLIP) to generate visual embeddings. This would allow detecting AI-generated faces (GAN artifacts) or inconsistencies between the profile picture and the bio text.
- **Time Series:** Analyze tweet timestamps as a time series to detect mechanical periodicity (e.g. posting exactly every hour).

Thank you for listening!

- GitHub repository: https://github.com/MaxKappa/botornot