Decoding Tweets: Unveiling Human vs. Machine Generation through Text Classification

Yu-Hsuan Lin
Northeastern University
Seattle, USA
lin.yu-h@northeastern.edu

Shibo Chen
Northeastern University
Seattle, USA
chen.shib@northeastern.edu

Yueyang Xu
Northeastern University
Seattle, USA
xu.yuey@northeastern.edu

Jessica James
Northeastern University
Seattle, USA
james.jess@northeastern.edu

Yu-Hsuan Lin, Shibo Chen, Yueyang Xu, and Jessica James. 2024. Decoding Tweets: Unveiling Human vs. Machine Generation through Text Classification. In . , 10 pages.

1 PROJECT DESCRIPTION

This project is to develop a robust model to distinguish between human-written text and machine-generated text. By focusing on tweets, we aim to explore the distinct characteristics and patterns in language, content, and style that differentiate human-generated content and machine-generated content. Our interest stems from the significant impact that fake news and misinformation can have on individuals in the context of social media platforms. As Twitter boasts a vast and diverse user population, this study contributes to the ongoing efforts to combat misinformation and enhance the overall trustworthiness of information shared on social media.

2 BACKGROUND SECTION

2.1 A Foundation Model Approach to detect Machine Generated Text

In the work done by Joanath [8],he illustrated cyber security issues raised by machine-generated texts, importance of distinguishing machine-generated texts from human-generated texts, and how to use a model to classify machine-generated texts and the results.

He used the Foundation Model framework approach, trained their own model using human-generated texts with self-supervised learning, and then fine-tuned the model using Transfer Learning approach on a small set of machine-generated texts.

Although the result is promising, further improvements are required to accommodate the growing LLM model.

2.2 Bot or Human? Detection of DeepFake Text with Semantic, Emoji, Sentiment and Linguistic Features

Alicia et al. also discuss the identification of critical linguistic[2], emoji, and sentiment characteristics to distinguish between machine-generated and human-generated posts on Twitter through a classifier.

The model includes BERT embeddings with semantic, emoji, sentiment and linguistic features, and multi-layer perception is used to incorporate additional features beyond BERT. And it is evaluated using logistic regression, support vector machines and random forest.

In conclusion, apparent differences between machine-generated texts and human-generated texts are found according to these features. The accuracy scores of the model reach 88.3%, with F1 scores ranging between 88.1% and 88.3%.

2.3 Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning 2023

Niful, Debopom, et al. explore the legal and ethical implications associated with the use of generative AI[5]. They emphasize the significance of differentiating between machine-generated and human-generated text and conduct experiments involving 11 algorithms to develop a model capable of distinguishing between the two.

The data underwent preprocessing utilizing TF-IDF vectorization for text and one-hot encoding for binary class columns, with a noteworthy observation that the removal of stop words negatively impacted classification performance, as the selection of stop words plays a crucial role in distinguishing human and AI-generated text. Subsequently, 11 algorithms, including Logistic Regression, Support Vector Machines, Decision Tree, K-Nearest Neighbor, Random Forest, AdaBoost, Bagging Classifier, Gradient Boosting, Multi-layer Perceptron, Long Short-Term Memory, and Extremely Randomized Trees, were employed for evaluation based on accuracy, recall, precision, F1-score, and Matthews Correlation Coefficient (MCC).

In conclusion, the Extremely Randomized Trees exhibit the highest F1-score at 76%, although the MCC score is 54%, still the highest among all algorithms, implying the potential for improvement with a more sophisticated model.

2.4 A Benchmark Dataset to Distinguish Human-Written and Machine-Generated Scientific Papers 2023

In the work done by Mohamed, Simon, et al [1], they assess the difficulty and importance of identifying genuine research in academic writing and scientific publication. They develop human-written and machine-generated scientific paper datasets with various generative AI and experiment with classifiers to build a model to detect authorship. The classifiers are then accessed in terms of generalization capabilities and explainability.

They created dataset including human-written, machine generated,using SCIgen, GPT-2, GPT-3, ChatGPT, and Galactica, and co-created scientific papers by humans and ChatGPT. They use a ChatGPT-based classifier without fine-tuning, alongside Large Language Model Feature Extractor, extracting explainability features using a Large Language Model and subsequently conducting classification using Random Forest.

In conclusion, Logistic Regression and Random Forest classifiers provide word-level insights; Galactica, RoBERTa, and GPT-3 reveal patterns in feature importance, indicating distinctions in language structures, and the Large Language Model Feature Extractor identifies abstract features such as grammar and syntax to understand complex relationships between words.

2.5 GLTR: Statistical Detection and Visualization of Generated Text

Gehrmann et al. used a statistical approach to detect differences in patterns between human and model generated texts[4].

With the assumption that AI systems tend to overgenerate from a limited subset of the true distribution of natural language, they developed a tool called GLTR (Giant Language model Test Room) which assesses each word in terms of its probability, absolute rank, and entropy of its prediction. From these results, GLTR then predicts the probability of the word being model generated, and visualizes the prediction by highlighting each word green, yellow, or red, in decreasing order of probability.

While GLTR significantly improved detection rate (54% to 72%) of generated text, the tool relies heavily on the assumption that models have a tendency for biased sampling. Models can employ different methods to evade detection by disrupting the said assumption, such as exclusively sampling from the tail of the distribution.

2.6 DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature

Mitchell et al. observed that large language model (LLM) generated texts tend to occupy negative curvature regions of the model's log probability function[7].

To distinguish between machine-generated and human written text, they introduced small perturbations to the text, and evaluated the difference between the log probability of the original text with the log probability of the perturbed sample. A sample is classified as model-generated if the difference in log probabilities exceeds a certain threshold.

Although DetectGPT achieved better performance in comparison to other zero-shot detection algorithms, it relies heavily on the availability of probability evaluation tools from the model, as well as a reliable and neutral perturbation function.

2.7 Automatic Detection of Machine Generated Text: A Critical Survey

The paper titled "Automatic Detection of Machine Generated Text: A Critical Survey" reviews the literature on detectors capable of distinguishing between text generated by Text Generative Models (TGMs) and human-written text.

Mitchell E. et al provide a comprehensive overview of the problem, highlighting the misuse of TGMs in creating authentic-looking fake news and reviews, and the efforts by the NLP and ML communities to develop accurate detectors for English[6].

They conclude by conducting an error analysis of state-of-the-art detectors and suggesting future research directions to improve detection capabilities.

2.8 Real or Fake Text?: Investigating Human Ability to Detect Boundaries between Human-Written and Machine-Generated Text

The paper " Real or Fake Text?: Investigating Human Ability to Detect Boundaries between Human-Written and Machine-Generated Text" investigates human ability to discern between human-written and machine-generated text within passages that start as human-written and transition to machine-generated text[3]. The authors employed a gamified platform, collecting over 21,000 annotations to analyze detection performance against variables like model size and genre. They concluded that while human detection varies, with training and incentives, annotator performance can improve, suggesting that textual genre and sentence-level features significantly influence detection accuracy.

3 DATASETS

As previously mentioned, this study aims to classify tweets into human-generated and machine-generated categories. To optimize our results, we have deliberately chosen tweets from three distinct topics: US Election 2020, FIFA World Cup 2022, and Game of Thrones S8. For the US Election Twitter dataset, it is important to note that we have strategically split the data to support both the Democratic Party and the Republican Party, ensuring an equal amount of representation for each, to maintain a balanced and comprehensive analysis. The selection of diverse topics not only facilitates the creation of a robust benchmark for comparing human-generated and machine-generated texts but also introduces variety into the classification task. By narrowing our focus to specific topics, we effectively mitigate noise in the dataset, enhancing its manageability and refining the signal-to-noise ratio for more effective training and evaluation processes. Furthermore, we have incorporated two additional datasets for model evaluation, focusing on previously unexplored topics: Health and Airline. These datasets are available on Kaggle: Mental Health, US airline

3.1 Human-Generated datasets

The datasets for the three topics can be found on Kaggle: Game of Thrones S8, US Election 2020, FIFA World Cup 2022.

The data preprocessing process begins by reading the relevant data from the original dataset and converting it into a DataFrame. The new DataFrame, exclusively contains the 'Tweet' column from the original dataset. Two additional columns, 'Topics' and 'human_generated', are introduced and initialized with appropriate values. The 'Topics' column is set to denote the specific topic associated with the tweets, while the 'human_generated' column indicates that these tweets are human-generated.

Text cleaning operations are then applied to the 'Tweet' column. Emojis and URLs are removed using a regular expression, and punctuation marks are stripped from the text. Special character sequences, such as '>' and '<', are replaced with an empty string. The count of profiles (mentions starting with '@') and count of URLs (starts with 'http') in each tweet is calculated, along with the count of emojis using the emoji.emoji_count function. A new column, 'clean_tweet,' is created by removing emojis, profiles, and URLs and retaining only alphabetical characters. All text is converted to lowercase for consistency.

Feature engineering is performed to create additional columns, such as 'count_emoji', 'count,' and 'count_url' storing the respective counts for each tweet, which will be used for further classification analysis. Two columns, 'Tweet_Length' and 'clean_tweet_ Length,' are introduced to capture the lengths of the original and cleaned tweets, respectively. The column order is adjusted for better organization.

Tokenization is then applied to the 'clean_tweet' column. The text is tokenized using the text_to_word_ sequence function. The tokenized lists are converted back to strings to facilitate lemmatization.

Due to the limited length of tweets and our subsequent vectorization using TF-IDF, we opted not to remove stopwords. Our experiments on a dataset of 3k samples showed that retaining stopwords actually led to higher accuracy. Consequently, we anticipate a decrease in performance if we remove stopwords for our larger dataset of 30k samples. Hence, we decided to retain stopwords in our text data.

Lemmatization is executed using the English language model in SpaCy, reducing words to their base or dictionary forms. The lemmatized strings are tokenized again, creating the 'tokenize_clean_tweet' column.

Finally, our primary objective is to develop an English language model. To achieve this, we're employing FastText, a language detection model trained on a diverse corpus that encompasses informal sources such as social media posts, forums, and chats. Our aim is to identify and select tweets written in English. Then, we randomly sample 10,000 rows for each topic to ensure a randomised selection process, reducing bias and ensuring that the selected samples are representative of the entire dataset.

The resulting dataset comprises 10 columns, namely: clean_tweet, Topics, Human_generated, count_emoji, count_profile, clean_tweet_ Length, Tweet_Length, Tweet, tokenize_clean_tweet, and token_length. This thorough data preprocessing process has been implemented to enhance the quality and uniformity of the tweet data, ensuring its suitability for subsequent analysis and classification tasks.

3.2 Machine-Generated datasets

The machine-generated tweets are created using the OpenAI library, wherein an API key is employed to initiate a request to the OpenAI API. This request is made to generate completions for a specified prompt, utilizing the powerful "gpt-3.5-turbo" model. A prompt example is shown below:

[Sample prompt: f"""Generate {max_tweets_per_request} tweets using the following prompt: You are a devoted supporter of Biden. Write an enthusiastic twitter post to show your support for him in the 2020 election. Do not exceed {character_limit} characters per tweet. Do not include line breaks within a tweet. """]

To efficiently manage the token limit for each request, we divide the generation task into multiple requests. Following the API call, we retrieve the response and extract the content of the generated completion using: response.choices[0].message.content.

To obtain high-quality machine data from each API call, we iteratively refined and engineered the prompt for text generation. Initially, we used OpenAI's GPT-3.5-turbo model with a broad prompt to encourage creativity and variety, allowing the model to explore different perspectives, sentiments, tones, and language patterns. However, this approach revealed significant disparities between human and machine-generated tweets across various dimensions: human tweets often lack full context, exhibit diverse sentiments, feature informal language with occasional spelling errors, and express varied tones ranging from casual to emphatic. In contrast, machine-generated tweets tend to provide complete narratives, convey predominantly positive sentiments, demonstrate flawless spelling, and adopt a more polite and formal tone.

This stark contrast posed a challenge as even basic models could achieve high accuracy levels of around 98% in distinguishing between the two types of tweets. To address this, we refined our approach to prompt engineering by incorporating actual human tweets as examples within the prompt itself. The revised prompt explicitly instructed the model to mimic not only the content but also the style, tone, and vocabulary usage of the provided human tweets. For instance, we required at least 60% of the vocabulary used in the generated tweets to be derived from the sample human tweets.

Upon this shift in prompt design strategy, the newly generated tweets resembled their human counterparts much more closely in terms of content and linguistic nuances. By emphasising the imitation of style, content, and tone from human tweets, our new prompt significantly elevated the challenge of distinguishing between human and machine-generated text, making it a more meaningful and nuanced task for our machine learning model.

4 METHODOLOGY AND RESULT ANALYSIS

4.1 Data analysis

In the Game of Thrones dataset, the average count of emojis per tweet is approximately 0.053, with a standard deviation of 0.52 and a maximum count of 26 emojis. The mean count of URLs is around 0.36, with a standard deviation of 0.58, and a maximum length of 3. The mean count of profiles in a tweet is roughly 0.41, with a standard deviation of 1.52 and a maximum count of 50. The mean token length is around 44.80, with a standard deviation of 5.67, and a maximum length of 67.

For the FIFA World Cup 2022 dataset, the mean count of emojis per tweet is approximately 0.86, with a standard deviation of 2.01 and a maximum count of 60 emojis. The mean count of URLs is around 0.49, with a standard deviation of 0.61, and a maximum length of 5. The mean count of profiles in a tweet is approximately 0.30, with a standard deviation of 0.85 and a maximum count of 25. The mean token length is around 18.71, with a standard deviation of 11.76, and a maximum length of 67.

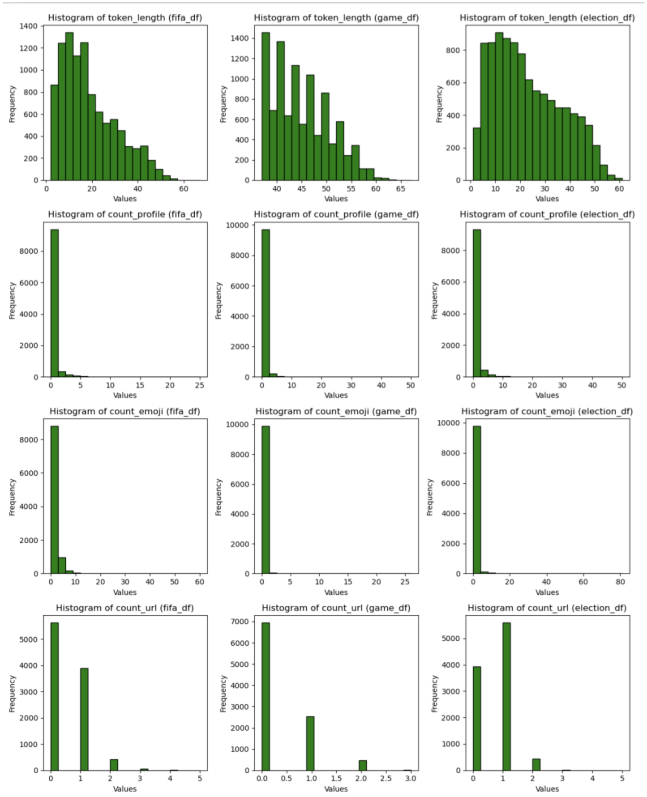

Figure 1: Distribution of Token Length, Profiles, Emojis, and URLs in Twitter Data

For the US Election 2020 dataset, the mean count of emojis per tweet is approximately 0.41, with a standard deviation of 1.93 and a maximum count of 81 emojis. The mean count of URLs is around 0.66, with a standard deviation of 0.58, and a maximum length of 5. The mean count of profiles in a tweet is about 0.76, with a standard deviation of 2.25 and a maximum count of 50. The mean token length is around 22.41, with a standard deviation of 13.65, and a maximum length of 61.

In Figure1, the plotted values serve as initial insights into the distribution and characteristics of these features, laying the groundwork for a more in-depth examination of how emoji usage and user mentions might relate to the classification of tweets as human-generated or otherwise.

Analysis of Sentiment, Readability, Personality Traits across Human and Machine Generated Datasets

This research incorporates a comprehensive approach to analyzing the textual data from FIFA, Game of Thrones and Election tweets, employing various natural language processing techniques to explore the sentiment, readability, personality traits, and morality dimensions. Sentiment Analysis: Using the TextBlob library, we calculated the sentiment polarity for each tweet, which helps in understanding the emotional tone—positive, neutral, or negative. The histogram and boxplot visualizations facilitate a comparative analysis between two distinct groups, potentially users and automated systems, highlighting the dominant emotional undertones in their tweets. Readability: The readability of the tweets was assessed using the Flesch-Kincaid grade level, provided by the textstat library. This measure indicates the academic grade level necessary to comprehend the text easily. The readability histograms offer insights into the complexity of language used by different groups, reflecting on their communication clarity and audience targeting. Personality traits were extracted using a pre-trained BERT model, specifically tailored for personality detection (Minej/bert-base-personality). This model was chosen due to its robustness and efficiency in handling natural language data, and its ability to infer personality traits such as Extroversion, Neuroticism, Agreeableness, Conscientiousness, and Openness from textual content. The extraction process involved the following steps: Tokenization: Tweets were tokenized using the BertTokenizer to convert text into a format suitable for model input. Model Prediction: The tokenized text was input into the BertForSequenceClassification model to obtain personality trait predictions. This step utilized a GPU-accelerated environment to enhance processing efficiency, especially pertinent when dealing with large datasets. Trait Scoring: The output logits from the model were transformed into a probability distribution representing the likelihood of each personality trait. These were then mapped back to their respective trait names for ease of interpretation.

Based on the visualizations for the FIFA, Game of Thrones, and election datasets, here are concise comparative analyses for user and machine-generated text across the three datasets:

FIFA Dataset:

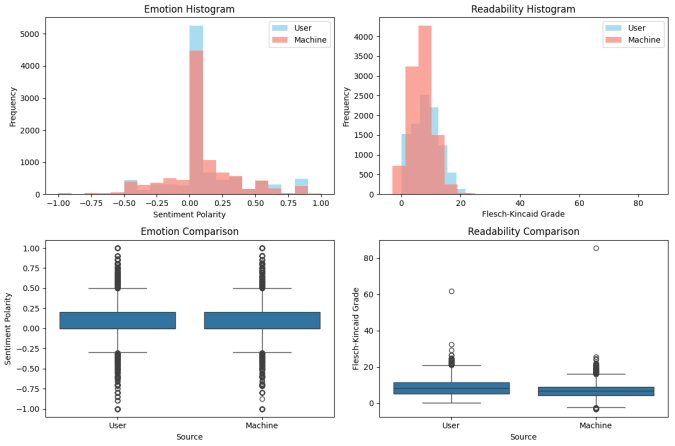

Figure 2: Sentiment and Readability Analysis

- **Emotion Analysis**: The emotion histogram and boxplot for the FIFA dataset show that user-generated content has a wider range of sentiment polarity, indicating a diverse emotional response. In contrast, machine-generated text shows a narrower sentiment polarity distribution, suggesting more uniform emotional content.
- **Readability Analysis**: The readability histogram reveals that user-generated tweets are generally more complex, as indicated by higher Flesch-Kincaid grade levels, compared

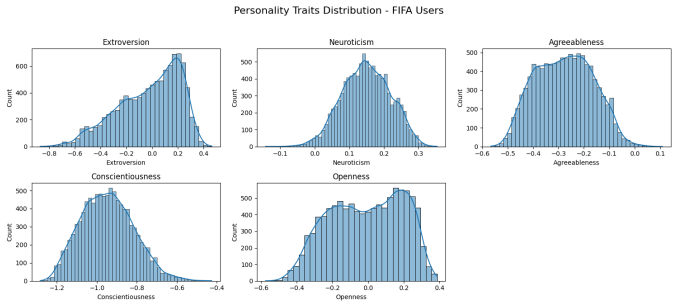

Figure 3: User generated Personality Analysis

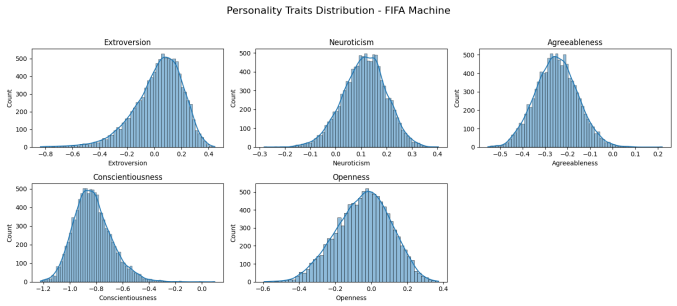

Figure 4: Machine generated Personality Analysis

to machine-generated text. This might imply that users engage with the topic using more nuanced language, while machine-generated content tends to be simpler and possibly more accessible.

- **Personality Traits**: The machine-generated data for the FIFA dataset shows distributions that are also normal and centered around zero for all traits. However, the 'Openness' trait has a slightly more irregular shape with what appears to be bimodal tendencies.User-generated data has distributions that are less smooth than the machine-generated ones, showing some irregularities. Notably, 'Conscientiousness' and 'Openness' traits show skewness towards the left, suggesting that a higher number of users score lower on these traits.

Game of Thrones Dataset:

- **Emotion Analysis**: The sentiment analysis depicts less variability in sentiment polarity among both user and machine-generated texts. However, there are notable extremes in sentiment polarity within user-generated content, reflecting strong emotional reactions to the series' content.
- **Readability Analysis**: Both user and machine-generated texts exhibit similar readability levels with a concentration around lower Flesch-Kincaid grades, indicating the text is relatively easy to read and comprehend. This could be due to the conversational nature of discussions about television series.
- **Personality Traits**: This machine-generated data displays normal distributions for all traits with peaks close to zero.

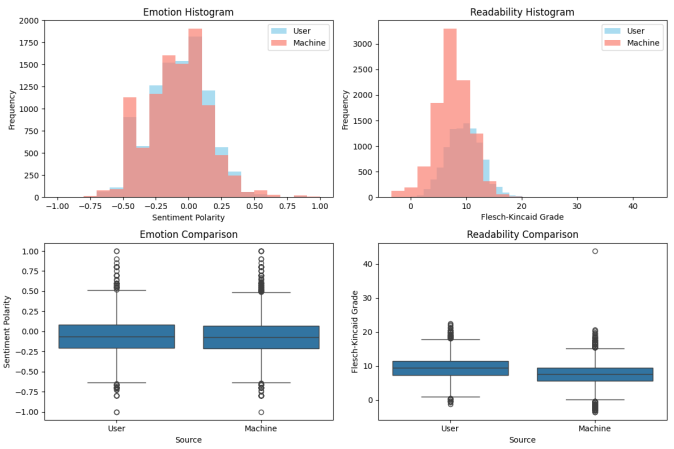

Figure 5: Sentiment and Readability Analysis

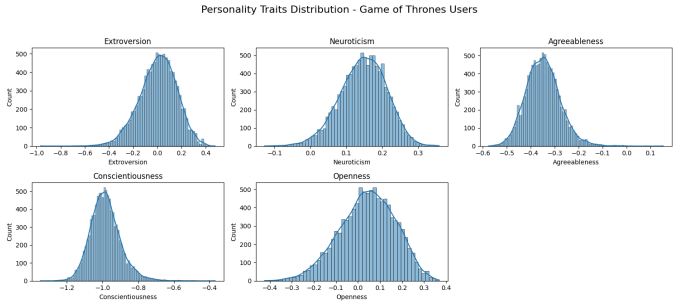

Figure 6: User generated Personality Analysis

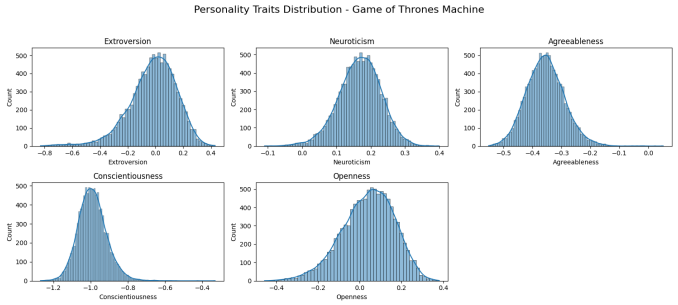

Figure 7: Machine generated Personality Analysis

Similar to the Election Machine data, it indicates a standardized range of trait scores.The user data shows normal distributions but with some skewness. Particularly, 'Extraversion' seems slightly skewed to the right, indicating a prevalence of higher extraversion scores among users.

Election Dataset:

- **Emotion Analysis**: The emotion comparison shows significant differences in sentiment polarity between user and machine-generated text. Users display a pronounced skew

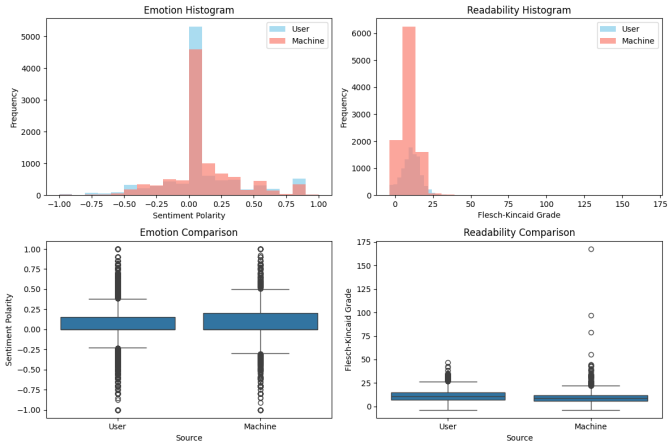

Figure 8: Sentiment and Readability Analysis

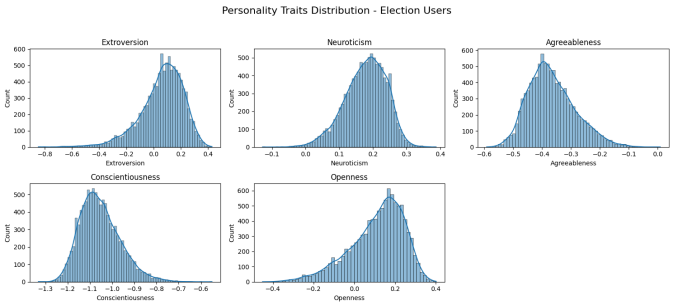

Figure 9: User generated Personality Analysis

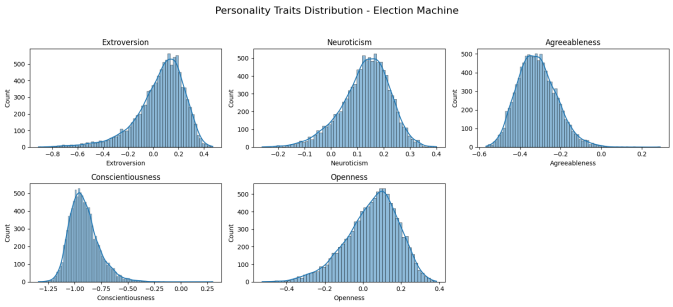

Figure 10: Machine generated Personality Analysis

towards one end of the sentiment spectrum, which could reflect passionate opinions during election periods. Machine-generated text shows more balance but less intensity in sentiment.

- **Readability Analysis**: The readability for the election dataset shows a broader spread in user-generated content, with some tweets reaching higher grade levels. This suggests that user discussions around elections can range from simplistic to more sophisticated discourse. Machine-generated content maintains a more consistent readability level, likely optimized for a general audience.
- **Personality Traits**: The machine-generated data for the Election dataset shows normal distributions for all traits, centered around zero, suggesting artificially generated or normalized data.The user-generated data for the Election dataset shows a similar pattern, with all traits exhibiting approximately normal distributions. This suggests that the user data may have been processed or that the sample is representative of a diverse population with a range of personality traits.

4.2 Model Analysis

Following the data analysis, further experimentation was conducted to refine the classification models and evaluate their performance. We extended our investigation to include additional features such as sentiment analysis scores and readability in tweet posting.

Model Experiments:

In this phase of experimentation, two sets of features were utilized: clean tweets, comprising solely of tokenized text, and profile mentions and emojis counts, offering supplementary contextual information. These feature sets were employed to train four classification models—Support Vector Machine (SVM), Random Forest, Decision Tree, and Logistic Regression—where each categorical feature was vectorized using TF-IDF Vectorizer. Moreover, to assess the performance of these models comprehensively, a KFold cross-validation approach was adopted with a fixed value of 5 folds across all models. The evaluation metrics employed included accuracy, precision, recall, and F1-score.

FIFA World Cup 2022 Datasets:

Table 1 illustrates the performance metrics of classification models trained on clean tokens from the FIFA dataset. It is evident that Logistic Regression achieved the highest accuracy at 76.44%, outperforming other models such as SVM and Random Forest, which attained accuracies of 75.00% and 72.74%, respectively. However, Decision Tree exhibited the lowest accuracy at 64.84%. Despite variations in accuracy, the precision, recall, and F1-score consistently portray high performance across all models. This suggests that the models are adept at correctly classifying instances from the clean tokens dataset, with Logistic Regression demonstrating superior accuracy while maintaining commendable precision, recall, and F1-score. The consistent high performance across metrics underscores the effectiveness of these models in handling the clean tokens data from the FIFA dataset.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	75.00	76.00	76.00	75.00
Random Forest	72.74	72.80	72.40	72.60
Decision Tree	64.84	65.50	65.30	65.30
Logistic Regression	76.44	76.90	76.20	76.40

Table 1: Performance Metrics of Classification Models for clean tweets (FIFA)

Table 2 presents the performance metrics of classification models trained on emoji profiles count from the FIFA dataset. These models exhibit notably lower accuracy compared to those trained on clean

tokens, with Decision Tree achieving the highest accuracy at 59.10%. The precision for all models is also lower, suggesting challenges in accurately identifying classes, particularly due to imbalanced data. Notably, Random Forest and Decision Tree show higher recall, indicating better identification of the majority class (non-emoji profiles) but struggle with the minority class (emoji profiles).

However, the trade-off between precision and recall varies across models. Random Forest and Decision Tree exhibit higher precision compared to SVM and Logistic Regression, implying a more conservative approach in identifying emoji profiles. The overall F1-score for all models remains lower compared to the clean tokens dataset, indicating the difficulty in effectively classifying emoji profiles based on their count. This suggests the necessity for further feature engineering or the utilization of more advanced models to enhance the performance of classification on emoji profile counts within the FIFA dataset.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	51.60	53.80	53.70	52.50
Random Forest	59.00	60.50	59.00	58.20
Decision Tree	59.10	60.40	60.00	59.10
Logistic Regression	53.67	55.60	53.50	50.10

Table 2: Performance Metrics of Classification Models for emojis and profiles count (FIFA)

Table 3 presents the performance metrics of classification models trained on a dataset consisting of clean tweets as well as counts of emojis and profiles.

Among the models, Support Vector Machine attained an accuracy of 60.40%, while Random Forest yielded the highest accuracy at 73.80%. Decision Tree showed a moderate accuracy of 65.10%, and Logistic Regression achieved the highest accuracy among all models, reaching 76.90%.

The precision, recall, and F1-score values for each model can be referenced directly from the table. Overall, the results suggest that incorporating counts of emojis and profiles alongside clean tweets could enhance the classification performance, with Logistic Regression showing the highest accuracy and F1-score among the models.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	60.40	61.70	60.70	61.30
Random Forest	73.80	73.90	73.70	73.80
Decision Tree	65.10	65.40	65.10	65.10
Logistic Regression	76.90	78.30	76.10	76.80

Table 3: Performance Metrics of Classification Models for clean tweets, emojis and profiles count (FIFA)

Game of Thrones S8 Datasets:

Table 4 showcases the performance metrics of classification models for clean tweets. Notably, Decision Tree achieved the highest accuracy at 99.98%, closely followed by Random Forest (99.96%), SVM (99.95%), and Logistic Regression (99.76%). Across all models, precision, recall, and F1-score remained consistently high, indicating robust performance in classifying clean tweets. These results suggest that the models effectively captured the underlying patterns within the text data from the Game of Thrones dataset, enabling accurate classification.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	99.95	100.00	100.00	100.00
Random Forest	99.96	100.00	100.00	100.00
Decision Tree	99.98	100.00	100.00	100.00
Logistic Regression	99.76	99.80	99.68	99.74

Table 4: Performance Metrics of Classification Models for clean tweets (Game of Thrones)

As for emojis and profiles count, all models exhibited notably lower accuracy compared to their performance with clean tweets, hovering around 51-54%. This suggests challenges in accurately classifying instances based solely on emojis and profiles count, indicating the complexity of extracting meaningful insights from such features. Despite the lower accuracy, precision, recall, and F1-score remained relatively consistent across models. This indicates that while the models struggled with this feature set, they still maintained a balanced performance in identifying patterns within the data.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	51.45	52.00	52.00	45.00
Random Forest	54.00	67.30	54.60	44.80
Decision Tree	53.99	67.50	44.40	42.50
Logistic Regression	53.78	61.10	49.76	44.18

Table 5: Performance Metrics of Classification Models for emojis and profiles count (Game of Thrones)

Table 6 presents the performance metrics of classification models trained on a dataset consisting of clean tweets, along with counts of emojis and profiles.

Support Vector Machine achieved an accuracy of 87.62%, exhibiting high precision, recall, and F1-score values, all around 87.60% to 87.80%. Random Forest, Decision Tree, and Logistic Regression models demonstrated exceptional performance, with accuracy scores of 99.98% for Random Forest and Decision Tree, and 99.66% for Logistic Regression. These models also achieved perfect precision, recall, and F1-score, indicating robust classification performance.

The results suggest that incorporating counts of emojis and profiles alongside clean tweets significantly enhances the classification accuracy of the models, especially evident in the Random Forest, Decision Tree, and Logistic Regression models, which achieved near-perfect accuracy and performance across all metrics.

US Election 2020 Datasets:

Table 7 outlines the performance metrics of these models when operating on clean tweets. Notably, Logistic Regression achieved the highest accuracy at 72.08%, closely trailed by SVM (71.57%), Random Forest (66.41%), and Decision Tree (59.74%). Although

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	87.62	87.80	87.80	87.60
Random Forest	99.98	100.00	100.00	100.00
Decision Tree	99.98	100.00	100.00	100.00
Logistic Regression	99.66	99.60	99.60	99.60

Table 6: Performance Metrics of Classification Models for clean tweets, emojis and profiles count (Game of Thrones)

there were variations in accuracy across models, their precision, recall, and F1-scores remained relatively consistent. This suggests a balanced performance in identifying tweets relevant to the US Election 2020, with SVM and Logistic Regression emerging as the top performers.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	71.57	71.80	71.40	71.60
Random Forest	66.41	66.58	66.58	66.57
Decision Tree	59.74	59.80	59.80	59.80
Logistic Regression	72.08	72.10	72.10	72.00

Table 7: Performance Metrics of Classification Models for clean tweets (US Election)

Table 8 delves into the performance of classification models utilizing counts of emojis/profile mentions. Here, the accuracy of all models was notably lower, ranging from 46% to 54%, compared to their performance with clean tweets. This indicates the inherent challenges associated with classifying tweets based solely on emoji and profile mention counts in the context of a politically charged event like the US Election 2020. Despite the lower accuracy, the models maintained relatively stable precision, recall, and F1-scores across the board.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	47.83	47.62	49.16	46.24
Random Forest	55.21	60.30	44.50	49.10
Decision Tree	55.16	61.96	42.34	47.04
Logistic Regression	53.58	54.80	54.40	52.50

Table 8: Performance Metrics of Classification Models for emojis and profiles count (US Election)

Table 9 showcases the performance metrics of classification models trained on a dataset containing clean tweets, along with counts of emojis and profiles. The result is similar to FIFA datasets.

Combined Datasets:

To ensure robustness and enhance predictive accuracy, the four models(Support Vector Machine, Random Forest, Decision Tree, Logistic Regression) are trained using a combined dataset comprising the three diverse datasets(FIFA, Game of Thrones S8, US Election 2020) with around 60,000 data entries. Subsequently, the performance of each model is rigorously assessed to determine the best-performing candidate.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	58.25	58.22	58.40	58.25
Random Forest	67.19	67.20	67.30	67.10
Decision Tree	60.67	60.46	60.53	60.46
Logistic Regression	72.38	74.80	69.40	72.50

Table 9: Performance Metrics of Classification Models for clean tweets, emojis and profiles counts (US Election)

As it is illustrated in Table 10, Logistic Regression performed the best among the models to distinguish machine-generated tweets from clean tweets, achieving the highest accuracy and balanced precision, recall, and F1-score values. However, there is still room for improvement in enhancing the performance of other models, particularly the Decision Tree model, to achieve comparable accuracy levels.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	79.71	80.15	79.52	79.68
Random Forest	79.26	79.06	79.26	79.13
Decision Tree	74.56	74.56	74.56	74.56
Logistic Regression	80.94	81.09	80.85	80.94

Table 10: Performance Metrics of Classification Models for clean tweets (Combined Datasets)

From Table 11, these results indicate that the models trained on emojis and profile counts alone did not perform as well as those trained on clean tweets. The accuracy values are relatively low, and there is a lack of balance between precision, recall, and F1-score across the models. Further analysis and feature engineering may be necessary to improve the performance of these models on this dataset.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	49.88	50.22	48.07	45.08
Random Forest	49.59	49.72	50.48	44.12
Decision Tree	49.62	49.62	50.08	48.60
Logistic Regression	49.54	59.42	44.09	37.82

Table 11: Performance Metrics of Classification Models for emojis and profiles count (Combined Datasets)

In Table 12, the findings indicate that the Support Vector Machine model demonstrated superior performance compared to others in this dataset, attaining higher accuracy and achieving a more balanced distribution across precision, recall, and F1-score metrics. Nonetheless, additional analysis and feature engineering may be warranted to enhance the performance of all models, notably Random Forest and Logistic Regression, on this amalgamated dataset encompassing clean tweets, emojis, and profile counts. Decision Tree model was omitted due to its prolonged runtime, and it was observed that clean tweets alone consistently outperformed the combined features of clean tweets, emoji count, and profile count.

Model	A (%)	P (%)	R (%)	F1 (%)
Support Vector Machine	66.54	66.46	66.46	66.50
Random Forest	49.75	49.74	50.02	49.46
Logistic Regression	49.41	49.10	49.10	49.40

Table 12: Performance Metrics of Classification Models for clean tweets, emojis and profiles counts (Combined Datasets)

4.3 Model Evaluation

Based on the performance metrics, it's evident that the Logistic Regression model excels when applied solely to clean tweets for combined datasets. Subsequent to the model selection phase, this trained Logistic Regression model was evaluated using two distinct datasets—one focused on mental health tweets and the other on US airline tweets—yielding valuable insights into its adaptability and effectiveness across diverse domains. Through this holistic methodology, our aim is to pinpoint a model that reliably predicts outcomes across various datasets.

Class	A (%)	P (%)	R (%)	F1 (%)
Machine-Generated	63.00	63.00	71.20	66.80
Human-Generated	67.00	67.00	58.60	62.40

Table 13: Performance Metrics of Classification Models for clean tweets (Mental Health)

For mental health tweets, the Logistic Regression model achieves an accuracy of 63.00% for machine-generated content and 67.00% for human-generated content. The precision scores for both classes remain consistent at 63.00% and 67.00%, respectively. However, there is a notable difference in recall rates, with the model demonstrating higher recall for machine-generated tweets (71.20%) compared to human-generated ones (58.60%). Consequently, the F1-scores stand at 66.80% and 62.40%, respectively, reflecting the model's effectiveness in discerning between machine-generated and human-generated mental health-related tweets.

Transitioning to the US airline tweets dataset, the Logistic Regression model continues to showcase its robust performance. Both classes, machine-generated and human-generated, exhibit similar accuracy rates of 69.00% and 68.00%, respectively. Moreover, precision scores remain consistent at 69.25% and 68.00%, while the model achieves higher recall for human-generated tweets (70.75%) compared to machine-generated ones (66.75%). Consequently, the F1-scores for the two classes are 68.00% and 69.00%, respectively.

Despite the differences in the datasets' content, the Logistic Regression model demonstrates consistent and reliable performance, underscoring its adaptability and effectiveness across diverse domains.

Class	A (%)	P (%)	R (%)	F1 (%)
Machine-Generated	69.00	69.25	66.75	68.00
Human-Generated	68.00	68.00	70.75	69.00

Table 14: Performance Metrics of Classification Models for clean tweets (US Airlines)

5 CONCLUSION

In our text preprocessing experiment conducted on sample datasets of size 6000, we observed that removing stopwords resulted in accuracy ranging from 60% to 80%. Conversely, retaining stopwords yielded an accuracy of 80%. This led us to conclude that stopwords contribute to distinguishing between human and machine-generated text. Therefore, we opted to retain stopwords in our text data preprocessing pipeline.

The visual analysis of sentiment, readability, and personality traits from user-generated and machine-generated data across different contexts—elections, FIFA, and Game of Thrones—reveals distinct patterns. User-generated content shows a wider variance in sentiment and readability, suggesting a more diverse range of expressions and complexity in language, potentially reflecting the emotional and cognitive investment of humans in these topics. The personality trait distributions are also broader, indicating a richer spectrum of personality expressions.

Our model evaluation underscores the remarkable performance of the Logistic Regression model in discerning between human-generated and machine-generated tweets, especially when applied to tokenized tweets excluding emojis and profile mentions. The experiments reveal that neither emojis nor profile mentions serve as strong indicators for distinguishing between the two, yielding only around 50% accuracy. Furthermore, employing a combination of tokenized tweets, emojis count, and profile mention count as features generally results in lower or similar accuracy compared to using tokenized tweets alone.

Across diverse domains, including a merged dataset comprising FIFA 2022, Game of Thrones Season 8, and US 2020 election tweets, the model consistently demonstrates its effectiveness. Achieving an accuracy of approximately 65% on new datasets covering mental health and US airline topics, our findings underscore the model's generalization capabilities and its potential for robust application across various contexts.

6 FUTURE DIRECTION

To optimize our current models and potentially enhance their performance, we plan to explore the application of deep learning techniques. These advanced techniques, such as recurrent neural networks (RNNs) or transformer-based architectures like BERT, have shown remarkable capabilities in capturing complex patterns and relationships in text data. By leveraging these methods, we aim to extract more nuanced features from the data, which could improve our models' ability to accurately classify tweets across different topics.

Moving forward, we aim to investigate why the Game of Thrones dataset shows much higher accuracy compared to the others. This investigation may involve a detailed analysis of the dataset's characteristics, including the distribution of tweet topics, the presence of specific keywords or phrases, and the engagement levels of users discussing the topic. By understanding these factors, we can potentially identify insights that contribute to the superior performance of our models on this dataset.

REFERENCES

[1] Mohamed Hesham Ibrahim Abdalla, Simon Malberg, Daryna Dementieva, Edoardo Mosca, and Georg Groh. 2023. A benchmark dataset to distinguish human-written and machine-generated scientific papers. *Information*, 14, 10, 522. DOI: 10.3390/info14100522.

[2] A. T. Y. Chong, H. N. Chua, M. B. Jasser, and R. T. K. Wong. 2023. Bot or human? detection of deepfake text with semantic, emoji, sentiment and linguistic features. In *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)*. Shah Alam, Malaysia, 205–210. DOI: 10.1109/ICSET59111.2023.10295100.

[3] Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: investigating human ability to detect boundaries between human-written and machine-generated text. 37, 11, 12763–12771.

[4] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. Gltr: statistical detection and visualization of generated text. *arXiv*. arXiv: 1906.04043.

[5] Niful Islam, Debopom Sutradhar, Humaira Noor, Jarin Tasnim Raya, Monowara Tabassum Maisha, and Dewan Md Farid. 2023. Distinguishing human generated text from chatgpt generated text using machine. *arXiv*. arXiv: 2306.01761.

[6] Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. Lakshmanan. 2020. Automatic detection of machine generated text: a critical survey. *arXiv*. arXiv: 2011.01314.

[7] Edward Mitchell, Yilun Lee, Alexander Khazatsky, Christopher Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. *arXiv*. arXiv: 2301.11305.

[8] Jonathan Pan. 2023. A foundation model approach to detect machine generated text. In number 2. Vol. 54. Chiang Mai, Thailand, (Apr. 2023), 405–408. DOI: 10.1109/TENCON58879.2023.10322333.