Team cnlp-nits-pp at PAN: Leveraging BERT for Accurate Authorship Verification: A Novel Approach to Textual Attribution

Notebook for the PAN Lab at CLEF 2024

Annepaka Yadagiri[1,*], Dimpal Kalita[1], Abhishek Ranjan[1], Ashish Kumar Bostan[1], Parthib Toppo[1] and Partha Pakray[1]

[1] *Computer Science & Engineering, National Institute of Technology, Silchar, Assam, India*

Abstract

The launch of ai-generated tools has attracted a lot of interest from the academic and business worlds. Effectively handling a broad spectrum of human inquiries, ai-generated tools offer clear, thorough responses that far outperform earlier open-source chatbots regarding security and use. People are interested in learning how powerful AI is and how far it has come from human specialists. However, concerns about the possible detrimental effects large language models like ChatGPT can have on society—including fake news, plagiarism, and social security problems—are beginning to surface. In this work, The dataset is provided from CLEF PAN-24 human-written text data and 13 different types of ai-generated models text data like alpaca-7b,bigscience-bloomz-7b1, chavinlo-alpaca-13b, Gemini-pro, gpt-3.5-turbo-0125,gpt-4-turbo-preview,meta-llama-llama-2-7b-chat-hf,meta-llama-llama-2-70b-chat-hf,mistralai-mistral-7b-instruct-v0.2,mistralai-mixtral-8x7b-instruct-v0.1,qwen-qwen1.5-72b-chat-8bit,text-bison-002,vicgalle-gpt2-open-instruct-v1. which approximately provides imbalanced data. The comparison of human-written and ai-generated data. We examine the features of ChatGPT's replies, the distinctions and shortcomings of human experts, and the prospects for LLMs based on the pan-24 dataset. We conducted extensive human assessments and linguistic examinations of ai-generated content compared to human content, yielding several intriguing findings. Then, we conduct in-depth research on the best ways to identify whether a given text was produced by ai-generated or humans. We construct three distinct detection systems, investigate critical variables affecting their performance, and test them in various contexts. Our solution approach for this task involves using the BERT model with a preprocessing model, where we achieved classification results with over 97.6% ROC-AUC for all the results included in this challenge.

Keywords

Large Language Models,, AI-Generated Content Detection, Natural Language Processing, Generative AI, BERT

1. Task

In cooperation with the Voight-Kampff Task at the ELOQUENT Lab, the Generative AI Authorship Verification Task at PAN uses a builder-breaker approach. ELOQUENT participants research innovative text creation and obfuscation techniques to evade detection, while PAN participants develop systems to distinguish between human and AI-generated content [1].

Detecting whether text is human or AI-generated is challenging due to several factors. First, AI-generated text from LLMs like GPT-4 is often highly coherent and contextually appropriate, making it difficult to distinguish from human writing. Additionally, LLMs can mimic human writing styles and nuances, further complicating detection. Statistical methods used to differentiate text, such as analyzing word frequency and sentence structure, often find significant overlap between human and

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

[*] You can use this document as the template for preparing your publication. We recommend using the latest version of the ceurart style.

[*] Corresponding author.

annepaka22rs@cse.nits.ac.in (A. Yadagiri); kalitadimpal112@gmail.com (D. Kalita); abhiran14082002@gmail.com (A. Ranjan); kumarashishbostan@gmail.com (A. K. Bostan); toppoparthib@gmail.com (P. Toppo); partha@cse.nits.ac.in (P. Pakray)

0000-0003-1326-1448 (A. Yadagiri); 0009-0008-0054-7459 (D. Kalita); 0009-0000-3721-430X (A. Ranjan); 0009-0007-0286-5162 (A. K. Bostan); 0009-0004-9756-1951 (P. Toppo); 0000-0003-3834-5154 (P. Pakray)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

AI text. Moreover, detection models trained on specific text types may not perform well on others, requiring extensive retraining and resources. Ethical and practical concerns also arise, such as the risk of false positives and negatives, privacy issues in data analysis, and the ongoing need to adapt to new AI techniques. Addressing these issues involves continuous advancements in detection algorithms and comprehensive research efforts.

2. Dataset Description

This section outlines the classification methods and specific model training approaches, Section 3.2 discusses the model's overall structure and Section 3.3 focuses on the key points of model training. The dataset, acquired via CLEF 2024 PAN [1], consists of about 1,087 rows of text composed by humans and approximately 14,131 rows of text produced by AI. The text comprises a combination of authentic and fraudulent news stories from different 2021 U.S. news headlines. Initially, the dataset contained numerous JSON encodings, which were removed in the first step. During further analysis of the cleaned dataset, NAN values were identified. These were addressed by consolidating all data into a single data frame. Using linguistic analysis, the text column extracted features such as average line length, vocabulary, word density, and POS tags. This provides an overview of the data processing steps, as shown in Figure 1. From this dataset, we extracted feature statistics. Table 1 represents statistics and feature extraction data.

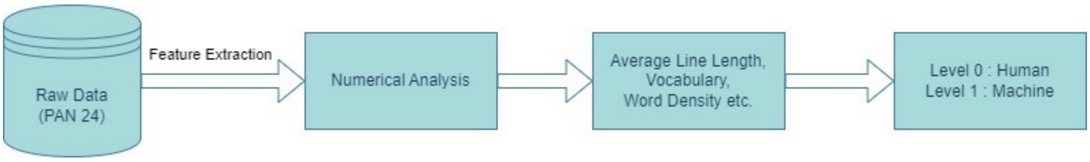

Figure 1: Data Processing steps

Table 1
The following table represents pan-24 dataset statistics and feature extraction data.

Model	Dataset size	Word density	Vocab	Avg line length	NOUN	PUNC	VERB	ADP	DET	PRON	ADJ	AUX	ADV	CCONJ	PROPN	PART	SCONJ	NUM	X	INTJ
human	1087	4.56	29820	27.39	17.17	10.61	14.0	10.25	8.6	3.94	5.77	0.81	3.23	2.59	10.64	0.29	10.25	1.72	0.012	0.013
alpaca-7b	1087	4.79	14409	30.82	16.38	10.52	9.04	9.88	8.63	1.98	5.37	0.76	1.83	2.99	15.99	0.12	9.88	1.4	0.0045	0.0065
bigscience-bloomz-7b1	1087	3.32	13938	24.18	17.66	8.72	10.54	11.06	9.41	2.7	5.36	0.7	2.03	2.35	15.48	0.19	11.06	2.1	0.005	0.0011
chav-inlo-alpaca-13b	1087	5.60	14476	28.82	16.37	9.07	10.17	9.97	8.43	1.95	5.92	1	2.06	3.15	20.28	0.1	9.97	1.48	0.0009	0.0009
gemini-pro	1087	4.15	24347	22.33	17.97	9.41	11.17	9.37	8.37	1.95	5.92	0.71	2.05	2.77	11.67	0.12	9.37	1.85	0.001	0.0013
gpt-3.5-turbo-0125	1087	4.57	22776	28.92	20.51	8.61	12.48	11.37	9.51	2.36	6.83	0.46	2.14	2.58	10.61	0.13	11.37	0.95	0.001	0.0011
gpt-4-turbo-preview	1087	3.99	26175	28.26	20.56	9.94	12.02	10.35	9.56	2.81	6.91	0.47	2.26	2.6	11.37	0.17	10.35	0.9	0.0014	0.0013
metallama 2-7b	1087	3.88	21431	26.58	19.07	10.28	10.88	8.99	9.99	3.66	5.98	0.74	2.21	3.14	9.13	0.12	8.99	1.4	0.003	0.0013
metallama-270b	1087	3.52	22422	25.16	18.52	9.98	12.3	9.39	8.98	3.6	5.91	0.62	2.26	3.04	11.56	0.12	9.39	1.15	0.0064	0.0032
mistralai-mistral-7b	1087	3.62	25147	24.67	18.3	11.15	9.55	9.76	8.43	3.2	5.62	0.67	2.26	3.04	11.56	0.12	9.76	1.24	0.0064	0.0032
mistralai-mixtral-8x7b	1087	3.92	26549	26.28	19.14	10.98	10.02	9.35	9.56	3.67	5.62	0.62	2.22	3.06	11.56	0.12	9.35	1.24	0.006	0.0024
qwen-qwen1.5-72b	1087	4.38	32658	27.84	18.19	10.56	10.6	9.7	9.36	3.16	5.35	0.59	2.1	3.05	11.44	0.1	9.7	1.51	0.003	0.0022
text-bison-002	1087	3.98	25960	26.01	19.16	10.56	12.83	9.22	9.19	3.66	6.65	0.59	2.23	2.88	11.44	0.1	9.22	1.01	0.0016	0.0016
vicugalle-gpt2-open-instruct-v1	1087	2.53	16920	30.03	17.68	9.41	12.8	10.45	9.35	3.16	5.9	0.69	2.02	3	13.32	0.13	10.45	1.56	0.0045	0.0047

3. System Overview

This section examines the linguistic differences between human-written and AI-generated texts. Next, the performance of existing detection algorithms is assessed using the PAN-24 dataset [2]. Finally, the criteria used by deep learning-based detection methods are investigated.

3.1. Vocabulary Features

This section examines the vocabulary characteristics of the PAN-24 dataset. The study is focused on the word choices made by AI-generated text and humans when responding to identical queries. Given the diversity of texts written by humans and AI, these differences are analyzed during the statistical procedure. The following traits were computed: in addition to lexicon measure (V), which measures the total number of unique words used in all responses, and average length (L), which measures the average number of words in each text, an additional characteristic named word density (D) is proposed. Word density is determined by the formula D = 100 × V / (L × N), where N is the number of answers. Density quantifies the degree to which words are employed intensively in a text. For instance, if 1,000 words of the text are published but only 100 distinct words are used, the density is 100 × 100 / 1,000 = 10. The higher the density, the more different words are used in the same text length. [3].

Lexical analysis Within the domain of NLP, every word can be categorized into one of several lexical categories. The part-of-speech (POS) tagging task aims to identify each word's grammatical class within a given phrase. In this section, the lexical distributions of various AI-generated and human texts in the PAN-24 dataset are computed using the POS module in NLTK [4]. The data is then arranged according to lexical percentage. As illustrated in Figures 2 and 3, various parts of speech are displayed. Figures 4 and 5 present punctuation and adposition tags, respectively. Finally, Figures 6 and 7 show determiners and pronouns. The statistics for the top ten lexical categories are displayed. Nouns (NOUN) make up the largest proportion of all lexical categories, while punctuation (PUNCT), verbs (VERB), adpositions (ADP), adjectives (ADJ), and determiners (DET) constitute most of the remaining categories.

When comparing human-written texts to AI-generated texts, the following observations can be made:

AI-generated texts have higher proportions of nouns (NOUN), verbs (VERB), determiners (DET), adjectives (ADJ), auxiliaries (AUX), coordinating conjunctions (CCONJ), and particles (PART) than human-written texts. This suggests that the rich knowledge embedded in AI-generated texts offers a more varied vocabulary, enhancing their informativeness.

Human-written texts contain higher proportions of adverbs (ADV) and punctuation (PUNCT) than AI-generated texts. This indicates that humans prioritize structure, consistency, and logical flow, in which AI-generated texts are comparatively weaker.

3.2. Model

A BERT-based sequence classification [5] and transformer-based model designed to understand the context of a word in search queries. Unlike traditional models that process text sequentially (either left-to-right or right-to-left), BERT considers the entire sequence of words simultaneously. This bidirectional approach allows BERT to grasp the context of a word based on its surrounding words, leading to better performance in NLP tasks.

Key Features of BERT:

- **Bidirectional Training:** BERT uses a Transformer architecture that reads text bi-directionally. This helps the model understand the context of each word more comprehensively.
- **Pre-training and Fine-tuning:** BERT involves two main stages:
  - **Pre-training:** The model is trained on a large corpus of text, learning to predict missing words in sentences (Masked Language Model) and the next sentence (Next Sentence Prediction).
  - **Fine-tuning:** The pre-trained BERT model is then fine-tuned on tasks such as text classification, named entity recognition, or question answering using task-specific data.

Our team plans to extract features from the original dataset, including the text and numerical columns. Initially, this dataset was utilized as training data for 3 epochs to train a new model, referred to as model A, using BERT. BERT, an enhanced version of previous models, incorporates a more significant number of parameters, more extensive training data, and larger batch sizes. It is trained significantly more significantly than CNN-BILSTM, which takes considerably longer. This extensive training allows BERT representations to generalize more effectively to downstream tasks and deliver superior performance compared to other models. As a result, the BERT model demonstrates high accuracy and faster processing speeds, as illustrated in Figure. 8.

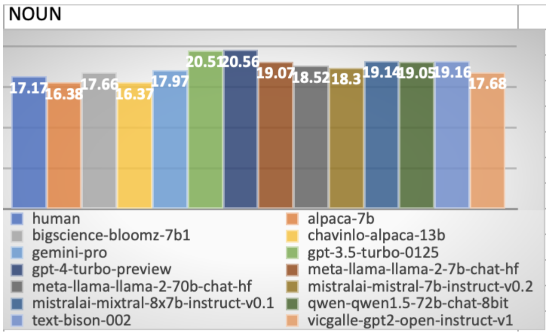

Figure 2: Pos tag Noun

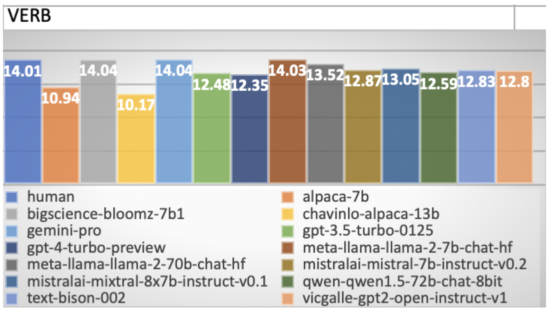

Figure 3: Pos tag Verb

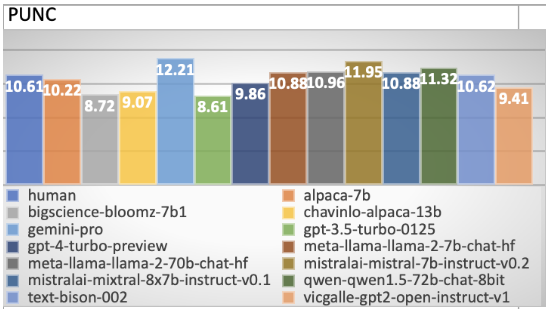

Figure 4: Pos tag Punctuation

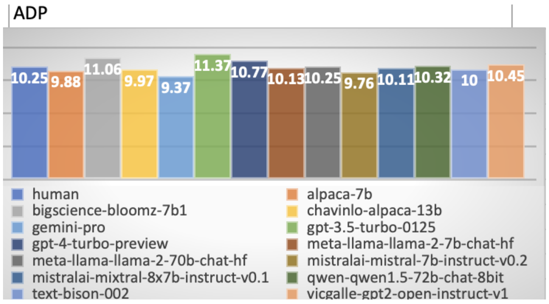

Figure 5: Pos tag Adposition

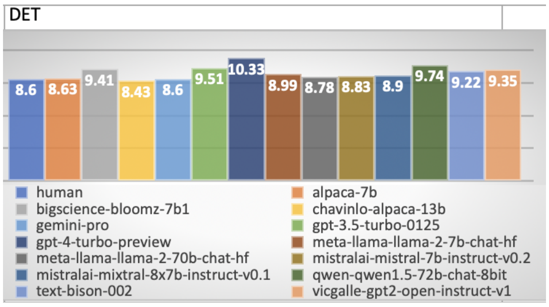

Figure 6: Pos Tag Determiner

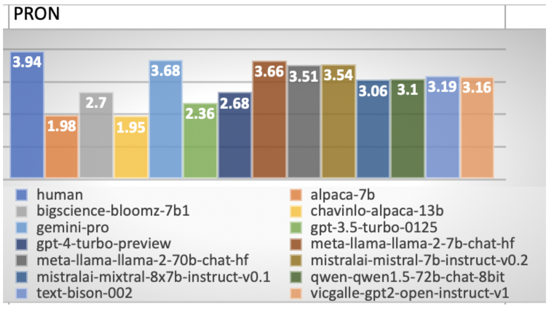

Figure 7: Pos tag Pronoun

3.3. Model Training

First, with Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz and NVIDIA A800-SXM4-80GB hardware platform, we will split the dataset into 80% training and 20% validation sets. Each training iteration will utilize a batch size of 32. The data consists of a text column along with numerical columns ([‘Vocabulary’, ‘Noun Count’, ‘Verb Count’, ‘AUX Count’, ‘NUM Count’, ‘PRON Count’, ‘ADV Count’, ‘INTJ Count’, ‘PART Count’]). This dataset will be input into the BERT model for sequence classification, incorporating the numerical features using PyTorch and Hugging Face Transformers.
The `CustomDataset` class inherits from `torch.utils.data.Dataset`.

- **__init__:** Initializes the dataset with text, numerical data, and labels, converting numerical data and labels to tensors.
- **__len__:** Returns the length of the dataset.
- **__getitem__:** Tokenizes text data, processes numerical features, and returns a dictionary with input IDs, attention mask, and label for a given index.

then Load a pre-trained BERT tokenizer and model from Hugging Face’s model hub. Create instances of the CustomDataset class for the training and validation sets. Create data loaders for the training

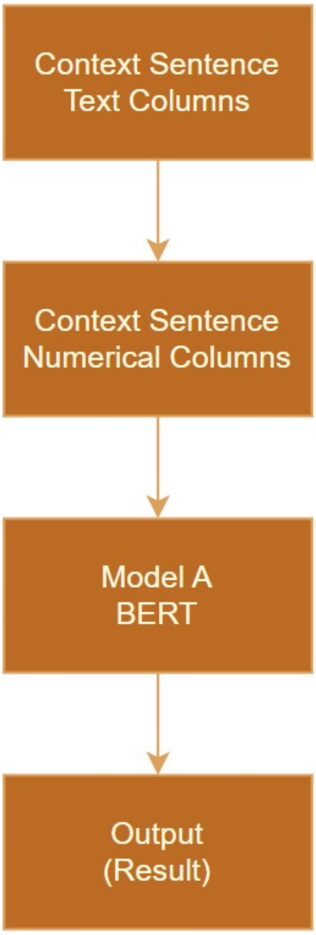

Figure 8: overall prediction process

and validation datasets with a batch size 32. AdamW: Optimizer with weight decay. CrossEntropyLoss: Loss function for classification tasks. Then, given to the model. tain() After 3 epochs of training, Model A is generated. Model A's training round takes about 20 minutes, while prediction time takes about 15 minutes. Then, based on the output results of Model A, our team has established the following criteria to evaluate the classification between the text and labels: In the data processing step, we are going to call the label data universally human '0' and ai-generated '1'. Based on the label data, we will predict which was a human-written text and which was ai-generated text. After training the model, we are going to check after training. We will take 20 percent of the data for testing purposes, whether it is predicted wrong or correct results. It will give good accuracy to the exact results that came. The model training process is shown in the figure 9.

3.3.1. Execution Steps

We have written software that can be run from the command line. An input file (an absolute path to the input JSONL file) and an output directory (an absolute path to the location where the results will be written) are the two arguments that the script requires.
We execute the command as follows in the terminal:

```
python3 model.py <input_file_path> <output_directory>
```

Here, `model.py` is the main Python file that loads and runs the model. The `<input_file_path>` is the path of the file containing the input texts, and the `<output_directory>` is the directory where the output file is saved.

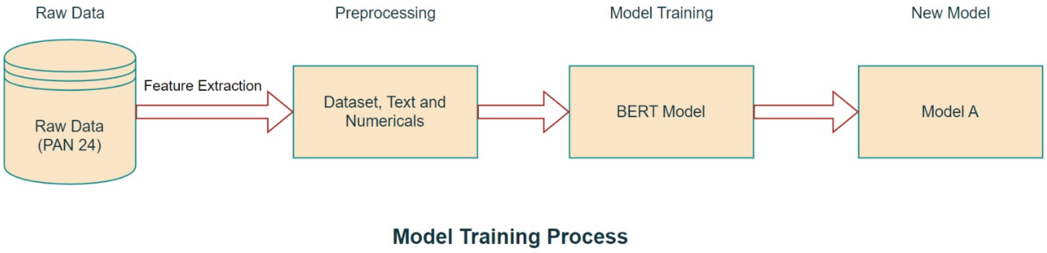

Figure 9: Model Training Process

3.4. Hyperparameters

The precise adjustments a user makes to control the learning process are known as hyperparameters. The best/optimal hyperparameters for learning algorithms must be selected during training to yield the most meaningful results. The hyperparameters used in our recommended techniques are shown in Table 2 we selected these values by analyzing the performance of the suggested methods with different combinations of hyperparameters.

Table 2
Hyperparameters that were applied to every experiment

Parameter	**Value**
Activation Function	Sigmoid
Optimizer	AdamW
Loss Function	nn_crossentropy
Learning Rate	2×10^{-5}
Batch Size	32
Number of Epochs	3
Dropout	0.2
ModelCheckpoint	Yes
EarlyStopping	Yes
Patience	5

3.5. Features Extracted

Feature extraction in NLP involves transforming raw text data into a structured representation that machine learning algorithms can use for various NLP tasks. The following features were extracted, and our model was trained on those parameters.

3.5.1. Average Line Length

In NLP, average line length is the mean number of characters or words per line in a text dataset like PAN-24. A sample text has been taken from this dataset. For example,

- Text: “President Joseph R. Biden Jr. calls for unity and a renewed commitment to democracy".
- Average characters per line: 74
- Average words per line: 12

3.5.2. Vocabulary

In NLP, vocabulary (vocab) refers to the set of unique words or tokens in a text dataset like PAN-24. A sample text has been taken from this dataset. For example,

- Text: “Biden’s inauguration is impacted by the pandemic and security threats."
- Vocabulary: "Biden’s," “inauguration," “is," “impacted," “by," “the," “pandemic," “and," “security," “threats"
- Size of vocabulary: 10

3.5.3. Word Density

In NLP, word density measures how many unique words (vocabulary) appear per unit of text, calculated as 100 times the vocabulary size divided by the product of the number of lines and the average line length.

A sample text has been taken from this dataset. For example, “A new chapter of American democracy begins amidst unprecedented times."

Step-by-Step Calculation:

- Vocabulary:
  - Unique words: "A," "new," "chapter," "of," "American," "democracy," "begins," "amidst," "unprecedented," "times"
  - Vocabulary size: 10
- Number of lines:
  - There is 1 line in the text.
- Average line length:
  - Line 1: "A new chapter of American democracy begins amidst unprecedented times." (70 characters)
  - Average line length: 70 characters
- Word Density Calculation: The word density (WD) can be calculated using the formula:

$$WD = \frac{100 \times \text{Vocabulary Size}}{\text{No of Lines} \times \text{Average Line Length}} \tag{1}$$

Where:

$$\begin{aligned} WD &: \text{Word Density} \\ \text{Vocabulary Size} &: \text{Number of unique words in the text} \\ \text{No of Lines} &: \text{Total number of lines in the text} \\ \text{Average Line Length} &: \text{Average number of characters per line} \end{aligned}$$

So, the word density of the text is approximately 14.29.

3.5.4. POS Tags

Part-of-speech (POS) tags are labels assigned to each word in a text to indicate its grammatical category, such as noun, verb, adjective, etc. POS tagging is a fundamental task in NLP that helps understand sentences’ syntactic structure and meaning. Explanation of POS Tags:

- Noun
  - Definition: Words representing people, places, things, or ideas.
  - Examples: “cat," “city," “happiness."
  - Usage: “The cat is sleeping."
- Verb
  - Definition: Words that describe actions, states, or occurrences.
  - Examples: “run," “is," “seem."

  - Usage: "She runs every morning."

- Punctuation
  - Definition: Symbols used to separate sentences and their elements and to clarify meaning.
  - Examples: ".", ",", "!"
  - Usage: "Hello, world!"
- Determiner
  - Definition: Determiners are words placed before nouns to specify quantity or definiteness.
  - Examples: "the," "a," "some."
  - Usage: "The apple is red."
- Pronoun
  - Definition: Pronouns are words that replace nouns.
  - Examples: "he," "they," "it."
  - Usage: "She loves her dog."
- Proper Noun
  - Definition: Proper nouns are specific names of people, places, or organizations.
  - Examples: "John," "Paris," "Google."
  - Usage: "Google is a search engine."
- Adjective
  - Definition: Adjectives are words that describe or modify nouns.
  - Examples: "happy," "blue," "tall."
  - Usage: "The tall building is new."
- Auxiliary Verb
  - Definition: Auxiliary verbs are used with main verbs to express tense, mood, or voice.
  - Examples: "is," "have," "will."
  - Usage: "She is running."
- Adverb
  - Definition: Adverbs modify verbs, adjectives, or other adverbs.
  - Examples: "quickly," "very," "well."
  - Usage: "He ran quickly."
- Particles
  - Definition: Particles are small words with grammatical functions that do not fit into other categories.
  - Examples: "to" (in "to go"), "not" (in "do not")
  - Usage: "She decided to go."
- Subordinating conjunctions
  - Definition: Subordinating conjunctions connect clauses to show a relationship between them.
  - Examples: "because," "although," "if"
  - Usage: "She stayed home because it was raining."
- Numerals
  - Definition: Numerals are words that represent numbers.
  - Examples: "one," "two," "third."
  - Usage: "She has two cats."
- X
  - Definition: Other categories of words that do not fit into the standard parts of speech.
  - Examples: Foreign words, typos
  - Usage: "She said ’ciao’ as she left."

3.6. Implementation

There are three major steps of our implementation as follows:

- **Tokenization and Model Loading:** This part sets up the tokenizer and the model into 19 distinct features as shown in Table 1. From among these features, only suitable features, The tokenizer and model configuration, are loaded from the 'bert-base-uncased' pre-trained model, and the actual model weights are loaded from a specified path. The model is set to evaluation mode and moved to the appropriate device (CPU or GPU).
- **TextDetector Class:** This class takes a text string as input tokenizes it, and then uses the model to get the logits (Logits are a neural network model's raw, unnormalized outputs). The logits are converted to probabilities using the softmax function. It assumes a binary classification model and returns the second class's probability (index 1).
- **Comparative Score Function:**

  ```
  comparative_score(score1, score2, epsilon=1e-3)
  ```

  This function compares two scores with a small threshold (epsilon) to avoid floating-point precision issues. It returns a value between 0 and 1 based on the comparison:

  - Returns a value between 0.5 and 1 if the first score is significantly higher.
  - Returns a value between 0 and 0.5 if the second score is significantly higher.
  - Returns 0.5 if the scores are very close (within epsilon).

  In the final function of calculating the result, it reads the line and parses it as JSON, then extracts the two texts (text1 and text2) and computes scores for both texts. It uses a comparative score function to determine a final score. Finally, the results are written in a JSONL file in the specified output directory.

4. Results

Table 3
Overview of the mean accuracy over 9 variants of the test set. We report the minimum, median, maximum, the 25th, and the 75th quantile, of the mean per the 9 datasets. **direct-velocity** is the name of the LLM model implemented in this paper. The submission scores 8th out of 30 on the PAN CLEF generated content analysis leaderboard.

Approach	Minimum	25-th Quantile	Median	75-th Quantile	Max
direct-velocity	**0.395**	**0.905**	**0.937**	**0.958**	**0.978**
Baseline Binoculars	0.342	0.818	0.844	0.965	0.996
Baseline Fast-DetectGPT (Mistral)	0.095	0.793	0.842	0.931	0.958
Baseline PPMd	0.270	0.546	0.750	0.770	0.863
Baseline Unmasking	0.250	0.662	0.696	0.697	0.762
Baseline Fast-DetectGPT	0.159	0.579	0.704	0.719	0.982
95-th quantile	0.863	0.971	0.978	0.990	1.000
75-th quantile	0.758	0.865	0.933	0.959	0.991
Median	0.605	0.645	0.875	0.889	0.936
25-th quantile	0.353	0.496	0.658	0.675	0.711
Min	0.015	0.038	0.231	0.244	0.252

4.1. Evaluation Metrics

Systems are assessed using the PAN authorship verification tasks as a benchmark. The metrics listed below are employed:

4.1.1. ROC-AUC

The region that falls within the receiver operating characteristic (ROC) curve. Characteristics of the Receiver Operating Area An indicator of the actual positive rate against the false positive rate at different threshold settings is the area under the receiver operating characteristic curve, or "Under the Curve." Higher numbers indicate better discrimination performance. It offers a total assessment of a model's capacity to distinguish between positive and negative classes.

4.1.2. Brier

The Brier score's complement (mean squared loss). For binary classification problems, the Brier score calculates the mean squared difference between the expected probability and the actual result (0 or 1). Lower Brier ratings indicate better calibration and accuracy of the probability predicted by the model.

4.1.3. C@1

A modified accuracy score that uses the average accuracy of the remaining instances to assign non-answers (score = 0.5). C@1 quantifies the percentage of cases in which the model's top-ranked prediction corresponds with the ground truth label. It is a typical assessment metric for recommendation or information retrieval systems.

4.1.4. F1

The harmonic mean between recall and accuracy. The F1 score is calculated by taking the harmonic mean of the two variables: recall, the ratio of accurate optimistic predictions to all real positives, and precision, which is the ratio of accurate optimistic predictions to all projected positives. Better performance is indicated by higher numbers, which strike a balance between recall and precision.

4.1.5. F0.5u

A precision-weighted F measure (modified F0.5 measure) that considers non-answers (score = 0.5) to be false negatives. While recall is less important than precision, the F0.5 score is comparable to the F1 score. When recall is less important than precision, like in situations where erroneous positives are more expensive than false negatives, it might be helpful.

4.1.6. Mean

The sum of all of the following measurements. The mean score indicates the average performance across all samples or occurrences in the evaluation dataset.

These metrics collectively provide insights into different aspects of model performance, including discrimination ability, calibration, accuracy, ranking quality, and the balance between precision and recall.

Table 3 shows the results, initially pre-filled with the official baselines provided by the PAN organizers.

4.2. Baseline Models

Baseline models are simple reference models used to establish a benchmark for evaluating the performance of more complex models in machine learning and natural language processing tasks. These models provide a standard or point of comparison, allowing researchers and practitioners to assess whether new models offer improvements in accuracy, efficiency, or other relevant metrics. By comparing against baseline models, it is possible to quantify the gains achieved by novel techniques and ensure that the advancements are meaningful and not merely coincidental. Six LLM detection baselines are used as references for the model results. These six LLM detection baselines are re-implementations from the original papers:

Table 4
Overview of the accuracy in detecting if a text is written by an human in task 4 on PAN 2024 (Voight-Kampff Generative AI Authorship Verification). We report ROC-AUC, Brier, C@1, F_1, $F_{0.5u}$ and their mean.**direct-velocity** is the name of the LLM model implemented from this paper. The submission scores 8th out of 30 on the PAN CLEF generated content analysis leaderboard

Approach	ROC-AUC	Brier	C@1	F_1	$F_{0.5u}$	Mean
direct-velocity	**0.976**	**0.877**	**0.959**	**0.934**	**0.94**	**0.937**
Baseline Binoculars	0.972	0.957	0.966	0.964	0.965	0.965
Baseline Fast-DetectGPT (Mistral)	0.876	0.8	0.886	0.883	0.883	0.866
Baseline PPMd	0.795	0.798	0.754	0.753	0.749	0.77
Baseline Unmasking	0.697	0.774	0.691	0.658	0.666	0.697
Baseline Fast-DetectGPT	0.668	0.776	0.695	0.69	0.691	0.704
95-th quantile	0.994	0.987	0.989	0.989	0.989	0.990
75-th quantile	0.969	0.925	0.950	0.933	0.939	0.941
Median	0.909	0.890	0.887	0.871	0.867	0.889
25-th quantile	0.701	0.768	0.683	0.657	0.670	0.689
Min	0.131	0.265	0.005	0.006	0.007	0.224

- Baseline Binoculars[6]
- Baseline DetectGPT [7]
- Baseline PPMd [8]
- Baseline Unmasking [9]
- Baseline Fast-DetectGPT [10]

5. Conclusion

This paper presents a bootstrap dataset of actual and false news items encompassing multiple 2021 U.S. news headlines, using the shared task on the PAN-24 dataset, which includes almost 1087 rows of human-written text. And different in almost 13 LLMs with 14181 rows(ai-generated text). Based on the PAN-24 dataset, we conduct broad considers counting human-written content assessments, phonetic investigation, and ai-generated content discovery tests. The human-written content assessments and phonetics analysis provide us with knowledge about the specific contrasts between human-written content and AI-generated text, which persuade our considerations of LLMs' future headings. The ai-generated content substance detection experiments outline a few imperative conclusions that can give advantageous guides to the research and improvement of AIGC-detection instruments. We make all our data, code, and models publicly available to facilitate related research and applications at our git hub repository AI vs Human

Acknowledgments

We express our gratitude to the National Institute of Technology Silchar's Department of Computer Science and Engineering and the Center for Natural Language Processing (CNLP) for providing the necessary infrastructure and assistance for this study.

References

[1] J. Bevendorff, M. Wiegmann, J. Karlgren, L. Dürlich, E. Gogoulou, A. Talman, E. Stamatatos, M. Potthast, B. Stein, Overview of the "Voight-Kampff" Generative AI Authorship Verification Task at PAN and ELOQUENT 2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. de Herrera

(Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[2] A. A. Ayele, N. Babakov, J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, D. Moskovskiy, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, N. Rizwan, P. Rosso, F. Schneider, A. Smirnova, E. Stamatatos, E. Stakovskii, B. Stein, M. Taulé, D. Ustalov, X. Wang, M. Wiegmann, S. M. Yimam, E. Zangerle, Overview of PAN 2024: Multi-Author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.

[3] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is chatgpt to human experts? comparison corpus, evaluation, and detection, arXiv preprint arXiv:2301.07597 (2023).

[4] S. Bird, Nltk: the natural language toolkit, in: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, 2006, pp. 69–72.

[5] C. A. C. Sáenz, K. Becker, Understanding stance classification of bert models: an attention-based framework, Knowledge and Information Systems 66 (2024) 419–451.

[6] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. URL: https://arxiv.org/abs/2401.12070. arXiv:2401.12070.

[7] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, Detectgpt: Zero-shot machine-generated text detection using probability curvature, 2023. URL: https://arxiv.org/abs/2301.11305. arXiv:2301.11305.

[8] D. Sculley, C. Brodley, Compression and machine learning: a new perspective on feature space vectors, in: Data Compression Conference (DCC'06), 2006, pp. 332–341. doi:10.1109/DCC.2006.13.

[9] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Generalizing unmasking for short texts, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 654–659. URL: https://aclanthology.org/N19-1068. doi:10.18653/v1/N19-1068.

[10] G. Bao, Y. Zhao, Z. Teng, L. Yang, Y. Zhang, Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature, 2024. URL: https://arxiv.org/abs/2310.05130. arXiv:2310.05130.