# AI-Writing Detection Using an Ensemble of Transformers and Stylometric Features

George Mikros[1], Athanasios Koursaris[2], Dimitrios Bilianos[3] and George Markopoulos[4]

[1]*Hamad Bin Khalifa University, LAS Building, Education City, Doha, Qatar*
[2]*National and Kapodistrian University of Athens, Department of Informatics and Telecommunication, Zografou, Greece*
[3]*National and Kapodistrian University of Athens, Department of Italian Language and Literature, Zografou, Greece*
[4]*National and Kapodistrian University of Athens, Department of Philology, Zografou, Greece*

Abstract

This study aims to develop an effective and precise methodology for detecting AI-generated text, leveraging the synergistic combination of transformer learning and stylometric features. The research utilized two datasets provided by the AuTexTification: Automated Text Identification shared task, a component of IberLEF 2023, the 5th Workshop on Iberian Languages Evaluation Forum held at the SEPLN 2023 Conference. Our team engaged in both English language subtasks, which included binary classification of texts as either human or AI-generated and multiclass classification to predict the specific AI writing model employed from a selection of six. Our main approach was to experiment with multiple Transformer models and, at the same time, to use an extensive stylometric feature engineering workflow. Each method (transformers and stylometric features) was first applied separately, and then we explored various ways to combine them. The most efficient method was based on ensemble learning utilizing majority voting employing the two most accurate transformer models in our training data and a comprehensive combined concatenation of many different stylometric feature groups. The macro-F1 scores on the test sets on subtasks 1 and 2 were 60.78 and 55.87, respectively, positioning our group above the median of the competing teams. This study underscores the potential of combining transformer learning and stylometric features to enhance the accuracy of AI-generated text detection.

**Keywords**

AI-writing detection, stylometry, transformers, ensemble learning

## 1. Introduction

During the last few years, the rise of Large Language Models (LLMs) has disrupted most Natural Language Understanding (NLU) and Language Generation (LG) tasks offering high-quality, coherent, and content-specific text [1]. In most cases, the output is indistinguishable from a human author's production. Moreover, most approaches for detecting whether AI has written a text or not have failed, or, at best, they show some promising results that, however, are not reliable enough to be used in real-world AI writing detection applications [2], [3], [4]. LLM writing detection has become even more challenging with the release of ChatGPT, the newest LLM from Open AI. ChatGPT is based on the GPT-3 and more recently on the GPT-4 LLM family of Open AI and can generate even more human-like and coherent text than its predecessors.

*IberLEF 2023, September 2023, Jaén, Spain*



This has led to a growing concern about the potential for malicious use of these models, such as creating fake news or impersonating individuals online.

Given the rapidly emergent malevolent uses, there is an urgent need to develop methods to discriminate AI vs. human written texts. For this reason, the AuTexTification[5]: Automated Text Identification shared Task was organized in the framework of the 5th Workshop on Iberian Languages Evaluation Forum at the SEPLN 2003 Conference[6]. The AuTexTification shared task provided two subtasks with related research objectives. In subtask 1, participants received a corpus, and they were asked to identify which texts were produced by an AI model or a human. To stimulate the models' ability to adapt to diverse writing styles, the training dataset in the specific subtask was based on three separate domains, while the testing datasets contained texts in two different domains. In subtask 2, participants were handed a series of texts, all of which were produced by six distinct AI models, each representing a stepwise increase in neural parameters of the generative model spanning from 2 billion to 175 billion. Both subtasks provided texts written in English and Spanish. However, our group focused on the English datasets and all experiments reported have been done in these language text collections.

In subtask 1, AI writing detection can be considered as a binary classification task with two classes (AI-generated vs. Human) which can be accommodated in the standard machine learning supervised classification paradigm. In this sense, this task is analogous to the authorship attribution task in which the AI and Human represent two different authors with separate stylometric profiles that can be distinguished due to their different frequency profiles of a common set of stylometric features. In subtask 2, each generative model has been pretrained with different neural parameters and produces linguistic output using a unique generative probabilistic method. For this reason, we can expect each generative model to produce distinct stylometric profiles and behave like a distinct author. Again, in this subtask, we can employ a standard authorship attribution model based on a multiclass supervised classification model.

The present paper is organized in five sections. In the next section (Section 2), we will delve into a literature review, focusing on the area of AI-generated text detection, while showcasing the latest studies in this field. Section 3 will outline the methodology we implemented in this shared task, detailing the various stylometric characteristics and machine learning techniques we utilized. In Section 4, we will highlight our range of experiments and the final iterations submitted for the shared task. Finally, in Section 5, we will explore the impact of our research, discuss some limitations, and consider potential future directions.

## 2. Literature Review

The recent advancements in automatic text generation facilitated by powerful language models have opened up new possibilities for research and development in various domains. Language models such as Generative Pre-trained Transformer (GPT) [7] , Pathways Language Model (PaLM) [8] and ChatGPT have demonstrated impressive capabilities in producing contextually coherent and fluent text [7].

Most of these models' architecture is transformer-based [9]. Models such as GPT-2 and CTRL have billions of parameters and they are trained on very large amounts of raw text from different sources (e.g., Wikipedia, Reddit). Such models can also be fine-tuned for a domain-specific task.

Text generative models have been utilized successfully in applications that range from code auto-completion to question-answering systems and even story generation [10]. However, the malicious use of these models to spread fake news, fake product reviews, and other misuses by adversaries (e.g., spamming, phishing) has become a pressing concern [11]. The task of guessing whether a text is produced by a human or a robot has become famous since the Turing test [12] but recently has emerged as a pressing AI safety task. Crucially, humans exhibit limited capability in accurately discerning fake news and comments generated by these models, performing no better than chance [11]. [13] show that human raters -even expert, trained ones- have consistently worse accuracy than automatic discriminators, while the performance gap is expected to grow as researchers train bigger and better models.

The approaches to address the task of identifying machine-generated text from human text mainly fall into three categories: Simple classifier approaches, zero-shot, and fine-tuning-based detection [14].

In simple classifier approaches, the classifier is trained from scratch. Some of these approaches employ classical machine learning methods. [4] use a simple baseline model with tf-idf unigram and bigram features on top of a logistic regression model. Then, the authors test a zero-shot approach. In zero-shot approaches, a pretrained model is employed to detect generated text from itself or similar models. [4] thus present a baseline consisting of a threshold on the total probability of the text sequence. The text is predicted as machine-generated if its likelihood, according to GPT-2, is closer to the mean likelihood over all machine-generated sequences than to the mean of human-generated sequences.

[15] provide GLTR (Giant Language Model Test Room), a tool of baseline statistical methods that can highlight the distributional differences in text generated by the GPT-2 model and human-written text. The generated text is sampled word by word from a next token distribution. The distribution usually differs from the one that humans subconsciously use when they write or speak [14].

The authors of GROVER [16], follow the fine-tuning based approach, using BERT, GPT-2, and GROVER as pre-trained language models. They find that GROVER outperforms the other models, concluding that the best models for generating disinformation are also the best at detecting their own generations. However, [4] found that fine-tuning a RoBERTa-based detector achieved higher accuracy on GPT-2 generated texts than fine-tuning a GPT-2 detector with equivalent capacity. This result might be due to the superior quality of the bidirectional representations employed by the RoBERTa language model compared to the unidirectional representation learning of GPT-2 [11]. RoBERTa-based detectors also performed better than other detectors in [17] and [14]. Finally, [18] experimented with several models: the Grover-based detector, GLTR, RoBERTa-based detector, and a simple ensemble that fused these detectors using logistic regression. The RoBERTa-based detector again outperformed the Grover-based detector and GLTR.

More recently [19] proposed a novel algorithm using stylometric signals to detect AI-generated tweets. The authors formulated two main tasks: firstly, detecting whether tweets in a timeline are human-written or AI-generated, and secondly, identifying the point in a mixed timeline where the authorship changes from human to AI. To achieve this, they proposed using stylometric features such as phraseology, punctuation, and linguistic diversity to capture writing style. These features are used in conjunction with pretrained language model embeddings. The

authors created an in-house dataset of human-written and AI-generated tweets for evaluation and also use an existing dataset called TweepFake. They tested two models: a fusion model that combines stylometric features and language model embeddings to detect human vs AI tweets, and a change point detection model (StyloCPA) using stylometric time series that detects when authorship changes from human to AI. The experiments showed that stylometric features improve the performance of pretrained language model-based detectors, especially when limited training data is available or the input text is short. The change point detection model also outperforms baselines in detecting the localization of human-AI authorship changes. The analysis showed that punctuation and phraseology features are most useful, while linguistic features require longer text to be effective models.

# 3. Methodology

## 3.1. Datasets

AuTexTification provided two English corpora (one for each subtask) for training purposes. Detailed descriptive statistics of both corpora per class can be found in Table 1 and Table 2 respectively:

**Table 1**
Descriptive statistics of the training corpora provided for subtask 1 per class.

| Descriptive Statistics | AI | Human | Total |
|---|---|---|---|
| N of texts | 16,798 | 17,043 | 33,841 |
| N of tokens | 930,016 | 896,829 | 1,826,845 |
| SD of N of tokens | 29.33 | 28.09 | 28.75 |
| Min N of tokens | 2 | 3 | 2 |
| Max N of tokens | 97 | 95 | 97 |

**Table 2**
Descriptive statistics of the training corpora provided for subtask 2 per class.

| Descriptive Statistics | A | B | C | D | E | F | Total |
|---|---|---|---|---|---|---|---|
| N of texts | 3,562 | 3,648 | 3,687 | 3,870 | 3,821 | 3,826 | 22,414 |
| N of tokens | 228,758 | 232,343 | 231,729 | 235,856 | 229,722 | 191,004 | 1,349,412 |
| SD of N of tokens | 25.97 | 25.93 | 26.48 | 23.96 | 24.45 | 24.85 | 25.73 |
| Min N of tokens | 2 | 2 | 2 | 3 | 2 | 2 | 2 |
| Max N of tokens | 97 | 96 | 97 | 97 | 97 | 94 | 97 |

After the training period, two separate unlabeled testing datasets (each for each subtask) were provided to be used for class prediction using the classification models which were developed in the training phase of the shared task. Descriptive statistics of both testing corpora can be found in Table 3 below:

**Table 3**
Descriptive statistics of the testing corpora for subtasks 1 and 2.

| Descriptive Statistics | Subtask 1 | Subtask 2 |
|---|---|---|
| N of texts | 21,832 | 5,605 |
| N of tokens | 1,370,209 | 335,897 |
| SD of N of tokens | 20.76 | 25.95 |
| Min N of tokens | 1 | 2 |
| Max N of tokens | 98 | 97 |

## 3.2. Features and classification algorithms

### 3.2.1. Transformers

Ever since the development of the BERT model by the team of Google Brain as well as the publication of the revolutionary "Attention Is All You Need" paper [9], the transformer architecture remains the backbone of some of the most prominent Deep Learning-based classification algorithms. The usage of the Attention Mechanism has given transformers a significant leverage against RNN-based architectures (LSTM, GRU), and, at the same time, Transfer Learning has allowed for the fine-tuning of already existing Transformer architectures in accordance with the data at hand. In order to obtain optimal results, it is also possible to pretrain certain architectures in various tasks, them being Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). For the purposes of the two different subtasks of the competition, a Transfer Learning approach was adopted, which involved the pretraining, customization and hyperparameter tuning of three different architectures:

- BERT: The original BERT model developed by Google, in its base-cased form [9], was the first model that was used for the purposes of the competition. The tokenization process allows for the conversion of words and subword units into an ID dictated by the vocabulary of the tokenizer. The IDs are then converted to 768-dimensional word embeddings based on the context. The texts are also padded, with padding itself being ignored due to the creation of an attention mask.
- RoBERTa: A more robustly optimized pretraining/training approach for BERT [20]. The roberta-base-openai-detector model was utilized, having been trained in a corpus of texts generated by the GPT and GPT-2 models. Again, tokenization is essentially identical, with the 768-dimensional word embeddings.
- ELECTRA: A "discriminator" approach in training transformers [21]. Pretraining involves replacing certain words with plausible alternatives to allow for the distinction between the original text and the reconstructed version. The contextual embeddings of the electra-base-discriminator model used for the task have a dimensionality of 1024.
- XLNet: Yet another different pretraining approach, this time favoring a permutation-based approach instead of predicting masked words in sequential order [22]. All words in a sentence can be predicted given the context of all other words in any given permutation. The xlnet-base-cased model used for the tasks at hand entails contextual embeddings not too dissimilar to those of the BERT transformer, with a dimensionality of 768.

### 3.2.2. Stylometric features

Over the past few decades, computational stylistics has developed into a robust and reliable field within Natural Language Processing (NLP). Its aim is to dissect an author's writing style into a variety of linguistic features. When combined, these features create a multi-faceted stylometric profile that uniquely correlates with the author's identity. The linguistic features that have been suggested as useful in authorship attribution tasks are numerous and cover all linguistic levels. Given that an author's style is a multi-layered phenomenon involving a complex and dynamic interaction of various linguistic levels, the most effective strategies tend to combine features that represent different and complementary facets of linguistic structure. For the needs of the shared task, we decided to use several feature groups that capture a wide range of linguistic phenomena and capture complementary linguistic functions. More specifically, we calculated the following feature groups in both training datasets:

- Author's Multilevel Ngram Profiles (AMNP): The Author's Multilevel N-gram Profile (AMNP), first proposed by [23] , and since then it has established its effectiveness in author attribution and profiling tasks, as supported by multiple studies [24] [25] [26] [27] [28]. AMNP is a method of representing a document that utilizes expanding n-gram sizes in both character and word units. This dual construction approach allows it to cover various linguistic levels. For the purposes of the shared study, we selected the 500 most frequent character and word 2-grams and the 500 most frequent words, yielding a total feature vector of 1,500 elements. The resulting vector constitutes the AMNP, a document representation that simultaneously captures sequences of both characters and words. All frequencies were converted to relative frequencies to avoid diverse text sizes bias.
- Stylometric and Language complexity indices (Stylo): In stylometric research, there is a long tradition of stylometric features that have been used for authorship attribution, including word and sentence length, lexical diversity indices, and a variety of features that capture quantitative properties of the text (e.g., distribution parameters of word frequencies, word entropy, etc.). Although these indices are not as efficient as the combination of word and n-gram frequencies, when they supplement other more robust document representations, they can increase the overall accuracy of any statistical model that predicts author identity. In this shared task we computed 74 stylometric features organized in the following wider thematic areas: a) Word and Sentence Lengths and standard deviations (measure in characters, words and syllables) b) Word Frequency distribution indices such as hapax legomena, dis legomena c) Quantitative aspects of text such as entropy, perplexity, and the corresponding standard deviations d) Lexical diversity indices such as various text size neutralized TTR measures (log-transformed TTR, root-transformed TTR, Mass TTR, Mean Segmental TTR, Moving Average TTR), Measure of Textual Lexical Diversity (MTLD), Hypergeometric Distribution D measure, functional diversity (ratio of content to functional words) e) Readability indices including Flesch-Kincaid reading ease and grade level, SMOG, Automated Readability Index, Dale-Chall, Linsear Write formula, Gunning-Fog score, Coleman-Liau index f) A large variety of metrics related to coherence, text quality, syntactic complexity and PoS tags frequencies. For more details about the indices involved in this feature group see [29].

- LIWC features (LIWC): LIWC (Linguistic Word Inquiry) is a dictionary-based tool which has been mainly used to measure aspects related to the psychological properties of a given text segment. It was created in the 1990s by James Pennebaker [30] as an attempt to automatically analyze essays and determine whether people who talk or write about their distressing personal experiences could show improvement in their physical and psychological health status. Although initially the LIWC tool has been primarily employed in psychological research, later it proved to be a successful "prediction tool" for several NLP (Natural Language Processing) tasks including authorship attribution [31] and author profiling tasks [32] [33]. Moreover, in recent experimentation [27] LIWC features were found to be the most efficient in predicting whether a text has been written by ChatGPT (GPT-3 model) or human outperforming even the GPT-3 embeddings. In this shared task we utilized the LIWC-22 tool [34] and its related English dictionary counting 120 linguistic features.
- GPT-2 word embeddings (GPT-2): Word Embeddings are amongst the most popular representation of a text's vocabulary. Words or phrases from the lexicon are transformed into vectors of real numbers and, subsequently, can be used in language modelling and feature learning approaches. These numerical vectors can capture text representations in an n-dimensional space, where words with the same meaning are represented similarly. This indicates that two comparable words are located extremely closely together in the vector space and thus have almost identical vector representations. Therefore, the objective of creating a word embedding space is to record some form of relationship in that space, be it a relationship based on meaning, morphology, context, or any other type. Since language produced by different authors exhibits systematic differences across all levels of its linguistic organization [23], we foresee that word embeddings can be used effectively for discriminating AI from Human writing. In this study we used the pretrained GPT-2 word embeddings [7] . OpenAI GPT-2 is a large autoregressive language model, built on a transformer-based architecture that encompasses 1.5 billion parameters. It was trained on a comprehensive dataset named WebText, composed of 8 million web pages, equating to 40 GB of internet text. The primary training objective of GPT-2 was to predict subsequent words in a sequence. Notably, GPT-2 demonstrated its remarkable capability for zero-shot task transfer, exhibiting proficiency in diverse tasks like machine translation and reading comprehension. For each text we calculated the average of word embeddings across the words it contains resulting in 768 dimensions for each text.

# 4. Results

## 4.1. Subtask 1 (English texts)

### 4.1.1. Experiments with Transformers

All experiments involving transformer-based classification algorithms were conducted using the PyTorch library, an open-source Deep Learning framework developed by Meta AI, allowing for the development of neural network architectures with an Object-Oriented Programming approach. PyTorch exhibits great interoperability with the Hugging Face Transformers library,

thus making it an excellent choice for Transfer Learning approaches.

BERT and RoBERTa models were initially pretrained by using MLM (Masked Language Modeling), where 15% of words in each sentence of every given text was masked, while the model attempted to restore all missing tokens, and NSP where given a sentence, the model made an attempt to guess the next sentence in the text. Experiments were made with and without pretraining the models, but the resulting models (bert-base-autextification and roberta-base-autextification) did not seem to impact the results in any meaningful way.

For the purposes of the binary classification task, performance was better when data preprocessing was minimal to nonexistent. Words and subword units were then converted to IDs, while padding to the maximum text length (depending on the vocabulary of the tokenizer and the subword units included) was applied. Padding was ignored during the training process due to the creation of an attention mask for every given entry.

The classifier architectures were created as classes, each with a constructor containing the various layers of the architecture, and a forward function for performing forward propagation. In every case, the first layer involved is that of the transformer architecture, followed by a dropout layer with a probability of 0.3, followed by a linear classification head with an input corresponding to the dimensionality of the transformer embeddings (e.g., 768 for BERT). The linear head input is taken either in the form of the pooled output (BERT/RoBERTa), the last timestep (ELECTRA), or the logits (XLNet). Since we are dealing with a binary classification task, the classification head is activated by a sigmoid function, converting the raw logits into a probability between 0 and 1 and being rounded. 0 represents generated texts, while 1 represents human-made ones.

Training was done in mini-batches with a batch size of 64 for the duration of 2 epochs total. Loss was calculated by the BCELoss function (Binary Cross-Entropy) provided by PyTorch, while the AdamW optimizer (Adam with weight decay) was used for optimization purposes, with an initial learning rate of 5e-5. Evaluation metrics were performed on a validation set (extracted by the train_test_split) function provided by the scikit-learn framework and comprising 20% of the original dataframe (all entries being random). Performance was evaluated by using Accuracy, Precision, Recall and F1-score. The ranking of the models in the subtask 1 dataset is displayed in Table 4:

**Table 4**
Comparison and Ranking of Transformer Models in subtask 1 dataset.

| Classifier | F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|
| ELECTRA | 0.934 | 0.934 | 0.934 | 0.934 |
| RoBERTa | 0.93 | 0.93 | 0.93 | 0.93 |
| BERT | 0.92 | 0.92 | 0.92 | 0.92 |
| XLNet | 0.89 | 0.89 | 0.89 | 0.89 |

The best results were yielded by the ELECTRA transformer, even though it is evident that transformer architectures seldom fall beneath the 0.9 threshold.

#### 4.1.2. Experiments with Stylometric Features

Experiments using the stylometric feature were performed using the PyCaret library [35] . PyCaret is a Python-based, open-source machine learning library that simplifies and automates machine learning workflows with its low-code approach. As an all-encompassing machine learning and model management tool, PyCaret accelerates the experimental process and fosters a versatile workspace conducive to efficient machine learning training. PyCaret compares several state-of-the art machine learning algorithms and provides a dashboard where the trained models are ranked by their classification metrics.

In the data preprocessing phase, we converted all values into z-scores to offset the impact of different scales across various features. Additionally, we eliminated any features demonstrating perfect collinearity to prevent bias in the training of multiple classification models. We then evaluated the models using a 5-fold cross-validation technique and ranking them according to their macro F1. Table 5 presents the ranking of models that were trained individually using each feature group, as well as a model trained using a concatenation of all features utilized in this study.

**Table 5**
Comparison and Ranking of Models Based on Individual and Combined Feature Groups in subtask 1.

| Feature Groups | Classifier | F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|---|
| All features + ELECTRA + RoBERTa | Ensemble | 0.95 | 0.95 | 0.96 | 0.95 |
| All features | XGBoost | 0.89 | 0.89 | 0.89 | 0.89 |
| GPT 2 | LightGBM | 0.86 | 0.86 | 0.86 | 0.86 |
| LIWC | CatBoost | 0.85 | 0.85 | 0.85 | 0.85 |
| Stylo | CatBoost | 0.84 | 0.84 | 0.83 | 0.85 |
| AMNP | LightGBM | 0.82 | 0.82 | 0.82 | 0.82 |

The highest F1 score (0.89) among the various feature groups is obtained by combining all the feature groups and using the XGBoost algorithm. The most single effective feature groups used are the GPT 2 embeddings and the LIWC features that yielded a F1 score of 0.86 and 0.85 respectively. To further enhance the classification efficiency of the employed models we experimented with various methods to combine the feature groups tabular datasets with the text only transformer models. We applied a process known as data prepending to our dataset. This involved appending a sequence of feature measurements to the end of each text item, with each measurement separated by the token [SEP]. To manage the resulting increase in sequence length, we utilized the Longformer and BigBird transformer models. These models are specifically designed to handle longer sequences compared to traditional transformer models and can accommodate a window size of 4092 dimensions. However, this method didn't work as expected and didn't increase the F1 in the training dataset. A much simpler approach was finally adopted using an ensemble method based on majority voting. We combined the predictions of the All-features group with the predictions with the best scoring transformers (ELECTRA and RoBERTa) and obtained the highest F1 (0.95) compared to all the other used methods. Given the best performance in the training data we decided to adopt it as one of our methods in the final run of the AuTexTification shared task. The final results in the testing dataset (Macro-F1: 60.78, CI: 60.14-61.49), justified our decision since our ensemble method scored better compared to

our other runs in both subtasks and got the 32nd position out of 76 runs.

## 4.2. Subtask 2 (English texts)

### 4.2.1. Experiments with Transformers

Procedures concerning the multilabel classification task were -to a degree- identical, with the main difference being the encoding of the given labels (A to F) into numbers (0 to 5), and the usage of the LogSoftmax function as activation for the linear classification head, as well as regular cross-entropy as a loss function (CrossEntropyLoss as given by PyTorch). For evaluating performance, the weighted average method of computing Accuracy, Precision, Recall and F1-score was preferred. The results can be seen in Table 6:

**Table 6**
Comparison and Ranking of Transformer Models in subtask 2 dataset.

| Classifier | F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|
| ELECTRA | 0.59 | 0.59 | 0.59 | 0.59 |
| RoBERTa | 0.58 | 0.58 | 0.57 | 0.58 |
| BERT | 0.57 | 0.57 | 0.57 | 0.58 |
| XLNet | 0.55 | 0.55 | 0.55 | 0.55 |

It seems that performance is not as optimal in subtask 2, with all AI-generated texts displaying a degree of similarity. Still, ELECTRA remains on top, if only marginally.

### 4.2.2. Experiments with Features

In subtask 2 we followed the same methodology discussed in subtask 1 and discussed in section 4.1.2. We used PyCaret library and utilized the 4 feature groups (GPT 2, LIWC, Stylo, AMNP) and their combined version. Table 7 presents the ranking of models that were trained individually using each feature group, as well as a model trained using a concatenation of all features utilized in this study.

**Table 7**
Comparison and Ranking of Models Based on Individual and Combined Feature Groups in subtask 2.

| Feature Groups | Classifier | F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|---|
| All features + ELECTRA + RoBERTa | Ensemble | 0.60 | 0.60 | 0.61 | 0.60 |
| All features | CatBoost | 0.53 | 0.54 | 0.53 | 0.54 |
| Stylo | CatBoost | 0.50 | 0.51 | 0.50 | 0.51 |
| AMNP | CatBoost | 0.45 | 0.45 | 0.45 | 0.45 |
| GPT 2 | LDA | 0.44 | 0.44 | 0.44 | 0.44 |
| LIWC | CatBoost | 0.44 | 0.44 | 0.43 | 0.44 |

The ranking reveals that the detection of different generative AI models is based on different features than the AI vs Human text classification. In this subtask, the most standard authorship attribution features, i.e., the stylometric features (Stylo) and the multilevel ngram profiles (AMNP) seem to work better and offer increased discriminatory power.

Again, in this subtask the best F1 was obtained through a simple majority voting between the two most accurate transformer models (ELECTRA and RoBERTa) and the All-features group. This ensemble approach was the best performing among our runs in the testing dataset of AuTexTification shared task obtaining the 21st position among 38 runs with Macro-F1 55.87 and CI ranging from 54.86 to 56.81.

## 5. Conclusions

This study explored various stylometric features and transformer-based models for detecting AI-generated texts. We experimented with different methods to improve classification accuracy, including preprocessing techniques, hyperparameter tuning, and ensemble methods. Our results demonstrate that combining multiple complementary approaches yields the most effective solution.

In the training dataset of the subtask 1, discriminating between human-written and AI-generated texts, our highest accuracy (95% F1 score) was achieved using an ensemble method based on majority voting that fused an XGBoost model trained on the combined features from our feature groups with predictions from the ELECTRA and RoBERTa transformer models. In subtask 2, identifying the specific AI model employed, an ensemble of the CatBoost model trained on the same combined features and the ELECTRA and RoBERTa transformer models achieved the top F1 score of 60%.

These findings highlight the benefits of fusing stylometric features and neural language models to detect AI writing. Stylometric features have a long history of effectiveness in authorship analysis and capture nuanced properties of writing style. In comparison, transformer models can detect more subtle patterns in the data. Combining these two approaches leverages their complementary strengths, confirming relative previous research [19].

Our study has some limitations. We only experimented with a subset of possible stylometric features and transformer architectures. Testing additional features and models may further enhance accuracy. Furthermore, our work focused on English texts; exploring other languages could reveal cross-linguistic patterns. Finally, continued progress in neural language generation will likely require continual advancement in detection techniques.

In conclusion, as concerns grow over misuse of AI writing, improved methods for discriminating machine- from human-generated text are increasingly important. Our work suggests that hybrid approaches uniting stylometric features and neural networks show promise for addressing this challenging problem. We hope our findings inspire continued progress in this critical area of AI safety.

## References


[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[2] L. Fröhling, A. Zubiaga, Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover, PeerJ Computer Science 7 (2021) e443.

[3] L. R. Varshney, N. S. Keskar, R. Socher, Limits of detecting text generated by large-scale language models, in: 2020 Information Theory and Applications Workshop (ITA), IEEE, 2020, pp. 1–5.
[4] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, et al., Release strategies and the social impacts of language models, arXiv preprint arXiv:1908.09203 (2019).
[5] A. M. Sarvazyan, J. Á. González, M. Franco Salvador, F. Rangel, B. Chulvi, P. Rosso, Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains, in: Procesamiento del Lenguaje Natural, Jaén, Spain, 2023.
[6] S. M. Jiménez-Zafra, F. Rangel, M. Montes-y Gómez, Overview of IberLEF 2023: Natural Language Processing Challenges for Spanish and other Iberian Languages, Procesamiento del Lenguaje Natural 71 (2023).
[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.
[8] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, N. Fiedel, Palm: Scaling language modeling with pathways, 2022. arXiv:2204.02311.
[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[10] A. Fan, M. Lewis, Y. Dauphin, Hierarchical neural story generation, arXiv preprint arXiv:1805.04833 (2018).
[11] G. Jawahar, M. Abdul-Mageed, L. V. Lakshmanan, Automatic detection of machine generated text: A critical survey, arXiv preprint arXiv:2011.01314 (2020).
[12] A. M. Turing, Computing machinery and intelligence, Mind 59 (1950) 433.
[13] D. Ippolito, D. Duckworth, C. Callison-Burch, D. Eck, Automatic detection of generated text is easiest when humans are fooled, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 1808–1822. URL: https://aclanthology.org/2020.acl-main.164. doi:10.18653/v1/2020.acl-main.164.
[14] T. Fagni, F. Falchi, M. Gambini, A. Martella, M. Tesconi, Tweepfake: About detecting deepfake tweets, Plos one 16 (2021) e0251415.
[15] S. Gehrmann, H. Strobelt, A. M. Rush, Gltr: Statistical detection and visualization of generated text, arXiv preprint arXiv:1906.04043 (2019).
[16] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against neural fake news, Advances in neural information processing systems 32 (2019).
[17] A. Uchendu, T. Le, K. Shu, D. Lee, Authorship attribution for neural text generation, in:

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 8384–8395.

[18] D. I. Adelani, H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi, I. Echizen, Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection, in: Advanced Information Networking and Applications: Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020), Springer, 2020, pp. 1341–1354.

[19] T. Kumarage, J. Garland, A. Bhattacharjee, K. Trapeznikov, S. Ruston, H. Liu, Stylometric detection of ai-generated text in twitter timelines, 2023. arXiv:2303.03697.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[21] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, arXiv preprint arXiv:2003.10555 (2020).

[22] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems 32 (2019).

[23] G. K. Mikros, K. Perifanos, Authorship attribution in greek tweets using author's multilevel n-gram profiles., in: AAAI Spring Symposium: Analyzing Microtext, 2013, pp. 17–23.

[24] G. K. Mikros, Authorship attribution and gender identification in greek blogs, Methods and Applications of Quantitative Linguistics 21 (2012) 21–32.

[25] G. K. Mikros, Systematic stylometric differences in men and women authors: A corpus-based study, in: R. Köhler, G. Altmann (Eds.), Issues in Quantitative Linguistics 3: Dedicated to Karl-Heinz Best on the Occasion of His 70th Birthday, number 13 in Studies in Quantitative Linguistics, RAM - Verlag, Lüdenscheid, 2013, pp. 206–223.

[26] G. K. Mikros, Blended authorship attribution: Unmasking elena ferrante combining different author profiling methods, in: A. Tuzzi, & MA Cortelazzo (2018). Drawing Elena Ferrante's Profile. Workshop Proceedings, Padova, 2017, pp. 85–95.

[27] G. K. Mikros, Detection of ai-generated texts and quantitative analysis of large language model outputs, in: Quantitative Linguistics Conference 2023 (QUALICO 2023), 2023. To be presented at the conference, June 28-30, 2023, Lausanne, Switzerland.

[28] M. A. Cortelazzo, G. K. Mikros, A. Tuzzi, Profiling elena ferrante: a look beyond novels, in: JADT 2018: Proceedings of the 14th International Conference on Statistical Analysis of Textual Data, 2018, pp. 165–173.

[29] L. Hansen, L. R. Olsen, K. Enevoldsen, Textdescriptives: A python package for calculating a large variety of metrics from text, Journal of Open Source Software 8 (2023) 5153.

[30] J. W. Pennebaker, Writing about emotional experiences as a therapeutic process, Psychological science 8 (1997) 162–166.

[31] J. Gaston, M. Narayanan, G. Dozier, D. L. Cothran, C. Arms-Chavez, M. Rossi, M. C. King, J. Xu, Authorship attribution via evolutionary hybridization of sentiment analysis, liwc, and topic modeling features, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2018, pp. 933–940.

[32] T. Isbister, L. Kaati, K. Cohen, Gender classification with data independent features in multiple languages, in: 2017 European Intelligence and Security Informatics Conference

(EISIC), IEEE, 2017, pp. 54–60.
[33] P. Panicheva, T. Litvinova, Matching liwc with russian thesauri: an exploratory study, in: Artificial Intelligence and Natural Language: 9th Conference, AINL 2020, Helsinki, Finland, October 7–9, 2020, Proceedings 9, Springer, 2020, pp. 181–195.
[34] R. L. Boyd, A. Ashokkumar, S. Seraj, J. W. Pennebaker, The development and psychometric properties of liwc-22, Austin, TX: University of Texas at Austin (2022) 1–47.
[35] Pycaret: An open source, low-code machine learning library in python, Online, 2023. Available at: https://www.pycaret.org.