Automatic Authorship Analysis in Human-AI Collaborative Writing

Aquia Richburg[1], **Calvin Bao**[2], **Marine Carpuat**[2]
University of Maryland
[1]AMSC, [2]Computer Science
{arichbu1, csbao, marine}@umd.edu

Abstract

As the quality of AI-generated text increases with the development of new Large Language Models, people use them to write in a variety of contexts. Human-AI collaborative writing poses a potential challenge for existing AI analysis techniques, which have been primarily tested either on human-written text only, or on samples independently generated by humans and AI. In this work, we investigate the extent to which existing AI detection and authorship analysis models can perform classification on data generated in human-AI collaborative writing sessions. Results show that, for AI text detection in the co-writing setting, classifiers based on authorship embeddings (Rivera-Soto et al., 2021) outperform classifiers used in prior work distinguishing AI vs. human text generated independently. However, these embeddings are not optimal for finer-grained authorship identification tasks: for authorship verification, n-gram based models are more robust to human-AI co-written text, and authorship attribution performance degrades compared to baselines that use human-written text only. Taken together, this suggests that the rise of human-AI co-written text will require adapting AI detection tools and authorship analysis techniques in the near future. We release our code at https://github.com/AARichburg/Human-AI_Authorship_Analysis.

Keywords: authorship analysis, human-AI collaboration, AI text detection

1. Introduction

With recent advances in NLP, people increasingly rely on Large Language Models (LLMs) when they write, whether for school assignments, daily emails, social media content, or professional writing endeavors such as news reporting and creative projects. In many of these settings, people do not only use LLM outputs verbatim. They might also edit them, or entirely ignore them, depending on how the resulting text is meant to be used, and also on their personal preferences. The introduction of LLM tools in the writing process thus blurs the boundary between human-written and AI-generated content. In the collaborative writing setting, people can edit AI-generated text and their writing can be influenced by LLM outputs. Furthermore, LLM prompts that lead to AI text generation might also become more personalized and contextualized than when AI text is independently generated, which might in turn impact output style.

Yet, existing research primarily focuses on distinguishing between human and AI-generated content generated independently (Gehrmann et al., 2019; Sadasivan et al., 2023; Mitchell et al., 2023; Krishna et al., 2023; Chakraborty et al., 2023), neglecting the nuances of different human writing styles, and ignoring co-authored text.

To address this gap, this work studies how the distinct properties of text co-authored by humans with AI support change systems' ability to distinguish (1) human-written vs. AI-generated text, and (2) individual human authors. We investigate whether we can automatically discriminate between human-written, AI-generated and human-edited AI-generated text and how well we can identify individual authors given human-written text only or a combination of human-written and AI-generated text.

We leverage the CoAuthor dataset (Lee et al., 2022), which records writing sessions where human participants respond to creative writing and argumentative writing prompts with the assistance of the GPT-3 LLM (Brown et al., 2020). We construct different views of the CoAuthor writing sessions based on how individual segments were generated, which allows us evaluate the injection of different types of co-written text in controlled settings.

We propose to use classification models inspired by the authorship analysis literature to capture more nuanced style differences than typical in classifiers designed for binary human vs. AI classification. Our findings show that this approach effectively differentiates between AI and human-written content within co-written documents. It also helps identify individual author styles in co-authored text more robustly.

This work makes the following contributions: (1) we construct datasets seeded from CoAuthor for evaluation of AI text detection and authorship analysis in the collaborative writing setting and (2) we demonstrate that existing models from the AI detection and authorship analysis literature can reasonably detect co-written text though there is a distributional shift from independently human- or AI-generated text.

LREC-COLING 2024, pages 1845–1855
20-25 May, 2024.

Prompt: All of the "#1 Dad" mugs in the world change to show the actual ranking of Dads suddenly.

Almost all dad's had one since over the past 6 months 2.1 billion of these mugs had been shipped out freely to almost every dad on earth. *The mugs were made by a company called "Mugs R Us" and were made to be a fun little gift to give to dads.* ... These traits were mostly gathered freely from social media sites but a significant portion was gathered through hacking and other illegal ways. ... The mugs were shipped out to every dad in the world and when they were turned on, they showed the ranking of the dad. ... *Some dads were so upset that they committed suicide, others vowed revenge on their neighbors.* ...

Table 1: An excerpt of a session in the CoAuthor data. Participants continue the story from the prompt. Each sentence is either generated by the human author (plain text), generated by GPT-3 (underlined) or generated by GPT-3 with edits from the human author (italics). GPT-3 text is generated conditioned on the previous text. Ellipses indicate content removed for space.

2. Background

This work draws from three distinct areas of the literature. By focusing on human-AI collaborative writing, it aligns with rapidly growing interest in designing LLM-based tools to support people's writing processes (Lee et al., 2022; Yuan et al., 2022; Jakesch et al., 2023; Laban et al., 2023), motivating such workshops as In2Writing (Huang et al., 2022) to coalesce interest across interdisciplinary fields towards the goal of building effective writing assistants. The potential for the prevalence of co-written text on the internet inspires this paper's analyses which examine how properties of the co-written text can be distinguished from either human-written or machine-written text. We create evaluation settings that complement past work on automatically detecting AI-generated text (Section 2.1) with authorship analysis models (Section 2.2).

2.1. AI Generated Text Detection

Methods for detecting AI-generated text have evolved in tandem with the capabilities of text generation models. Badaskar et al. (2008) exploit n-gram language model generation by utilizing n-gram features to distinguish real (original text) from fake (n-gram LM-generated) articles. Gehrmann et al. (2019) point to generation strategies that sample from the head of the distribution like max sampling (Gu et al., 2017) or beam search (Chorowski and Jaitly, 2016; Shao et al., 2017) that bias models to generating less diverse text; and visualize AI-generated text through token-level scoring such as the probability and entropy of tokens within a text segment. In response to the shift towards increasingly large LLMs, various neural detection systems have been introduced (Fabien et al., 2020; Abbasi et al., 2022; Mitchell et al., 2023), and a benchmark dataset (TuringBench (Uchendu et al., 2021)) has also been developed for controlled comparisons among systems. Dugan et al. (2022) tasked human participants with detecting the transition point between human- and AI-text within a document. Annotators overwhelmingly point to common sense or irrelevancy issues as rationales for deciding a segment is AI-generated. With continued improvements in the quality of generated text, some have argued that the AI- and human-generated text will become too similar to be distinguishable (Sadasivan et al., 2023), while others suggest that, in practice, the amount of text available is key to provide enough signal for detection (Chakraborty et al., 2023).

Overall, two family of techniques emerge from this literature: the first based on n-gram statistics, such as OpenAI's strong baseline built from a logistic regression classifier using Tf-idf-weighted n-gram vectors (Solaiman et al., 2019) and the second which simply consists in fine-tuning an LLM such as RoBERTa (Liu et al., 2019) for the detection task, as was done by Sadasivan et al. (2023); Chakraborty et al. (2023) and in the OpenAI GPT-2 detector (Solaiman et al., 2019). We will thus use these two classifiers as representative of AI-text detection in our study.

2.2. Authorship Analysis

Computational authorship analysis is a field of research that aims to automatically analyze authorship styles within written text. Computational authorship analysis systems motivate many real-world applications, including for plagiarism detection (Stamatatos and Koppel, 2011), detection of hacked accounts (Junior et al., 2017), and forensics (Yang and Chow, 2014; Ainsworth and Juola, 2019). These real-world needs, as well as the need to technically standardize and improve algorithms, models, and datasets, have motivated an annual shared task, PAN [1]. PAN covers ground spanning multiple tasks including profiling, style change detection, diarization, verification, attribution, and obfuscation.

[1] https://pan.webis.de/shared-tasks.html

We focus our analysis in this paper using methods from the **verification** (determining whether two texts were written by the same author) and **attribution** (identifying the correct author from a closed-set of authors, represented by their writing samples) tasks.

Verification Verification refers to the binary classification task where given two writing samples decide whether the texts are written by different or the same authors. The PAN2022 shared task on Authorship Verification showed that verification is difficult in settings with varied domains and text lengths. The overview (Stamatatos et al., 2022) showed that a naïve baseline based on character n-gram representations of document pairs was demonstratively as effective as more sophisticated, neural-based methods at distinguishing authors in the verification setting. We use this n-gram baseline model in our own experiments.

Attribution Attribution consists of identification models, and often is evaluated using a closed-world set of candidate authors with writing samples from which an identification model can draw meaningful features in order to classify unlabelled writing. Models have frequently been trained on these features, which can operate at the lexical-, syntactic-, semantic-, character-, and application-specific levels (Stamatatos, 2009). More recently, neural embeddings have proven to be effective at capturing authorship distinction, thanks to the availability of larger-scale labelled datasets that can be used to learn authorship representations. We use one such embedding model (Rivera-Soto et al., 2021) as a basis for validating our research questions.

3. Approach

This section describes our approach to studying authorship in human-AI co-written text, including the data (Section 3.1) and models (Section 3.2) shared across experiments.

3.1. CoAuthor Dataset

Our experiments rely on the CoAuthor Human-AI Collaborative Writing Dataset (Lee et al., 2022). This dataset consists of writing sessions conducted by human participants in response to prompts, with the assistance of a state-of-the-art LLM (GPT-3). Participants spend about 15 minutes writing in a session where they have the option of accepting, editing or ignoring text generated by GPT-3. Table 1 presents an excerpt of an author response to a prompt delineating the different text categories.

Prompts are divided into two categories of ten prompts each: argumentative essays and creative writing. An example of a creative and argumentative prompt is shown in Table 2. Authors were not required to write in all prompts and sessions were not required to have all text types. Within a CoAuthor writing session, the participant has the option to prompt GPT-3 for suggestions conditioned on the previous context of the session. If the participant does not prompt GPT-3, then we consider the text written next to be human-generated (`human`). If the participant elects to prompt GPT-3 and accepts its output without any modification, then we put the text in the GPT-generated category (`GPT`). If the participant elects to prompt GPT-3 but edits the GPT-3 output in any way, then the text falls in the edit category (`edit`). As a result, a CoAuthor session written by a participant in response to a prompt, can be viewed as a sequence of segments that belongs to one of the above three types.

In total, there are 1447 sessions, written by 62 authors. Each session has an average length of 24.4 segments. Overall, 66% of segments fall within the `human` category, 20% within `GPT`, and 14% within `edit`.

Selecting different subsets of segments lets us construct different views of this data to address two main research questions:

- Can we automatically discriminate between `human`-written, `GPT`-generated, and `edit` text? This is a binary or ternary classification class depending on the types included.

- How well can we automatically identify individual authors of a session when using `human`-written text only vs. a mixture of `human` and `GPT` text?

3.2. Classification Models

We consider four classification models including n-gram models and neural/embedding based models.

Logistic Regression We use standard logistic regression classifiers, based on one of two feature types: LUAR embeddings (Rivera-Soto et al., 2021) and Tf-idf-weighted character n-gram vectorization. The LUAR (Learning Universal Authorship Representations) embeddings are sentence embeddings trained for authorship identification on texts extracted from Reddit posts. n-gram based features have provided strong baselines for detecting AI-generated text in past work (Solaiman et al., 2019). Across experiments, we will use a maximum length of 200 tokens per input. We implement n-gram vectorization using the scikit-learn (Pedregosa et al., 2011) toolkit. We use character n-grams of size 4 with a vocabulary size of 3000.

Prompt code	Prompt text (source url)
dad	All of the "#1 Dad" mugs in the world change to show the actual ranking of Dads suddenly. (https://www.reddit.com/r/WritingPrompts/comments/6gl289/wp_all_of_the_1_dad_mugs_in_the_world_change_to/)
stereotype	What Stereotypical Characters Make You Cringe? What stereotypical characters in books, movies or television shows make you cringe and why? Would you ever not watch or read something because of its offensive portrayal of someone? (https://www.nytimes.com/2017/11/16/learning/what-stereotypical-characters-make-you-cringe.html)

Table 2: An example of a creative (dad) and argumentative (stereotype) writing prompt from the CoAuthor data. Participants continue writing a story from the initial creative prompt or respond to the question provided in an argumentative prompt. Prompts were sourced from the attached url and modified by Lee et al. (2022).

RoBERTa We also fine-tune a pre-trained RoBERTa (Liu et al., 2019) model for sequence classification. RoBERTa is commonly used as a foundation in fine-tuning models for AI-generated text detection (Solaiman et al., 2019; Chakraborty et al., 2023; Sadasivan et al., 2023). We initialize our model with `roberta-base`, consisting of Transformer blocks of 24 layers with 16 self-attention heads and size 1024 hidden dimension. We utilize the baseline parameters as we are interested in measuring RoBERTa's out-of-the-box capabilities in the collaborative writing scenario.

Character n-gram Distance For authorship verification, we additionally include a standard character n-gram model (CNG) class, released as part of the PAN2022 (Stamatatos et al., 2022) authorship verification task. As character n-grams are highly indicative of authorship style, the model first creates Tf-idf-weighted character n-gram vectors for each input text. After computing cosine similarities for each text pair, a grid search identifies an optimal verification threshold. Similarities are then re-scaled to compute pseudo-probabilities indicating the likelihood that the two texts come from the same author. We use the default settings of n-grams of length 4 and a vocabulary size of 3000. We run one iteration of grid search, and do not use bootstrapping.

4. AI Detection

We evaluate whether human-generated text in the CoAuthor data can be distinguished from AI-generated text in two settings: the binary `human` vs. `GPT` classification task which aligns the typical framing of AI detection in prior work, and the ternary `human` vs. `GPT` vs. `edit` classification task which reflects the different generation modes specific to the co-writing setting.

4.1. Experiment Setup

We construct data to train and evaluate four versions of the AI detection classification task. We consider the combination of the binary or ternary prediction tasks at segment-level or session-level granularities.

Each session is divided into its three text categories (possibly empty) and pooled into S_{H}, S_{G} and S_{E} for `human`, `GPT` and `edit` text, respectively. Segment-level examples are created by further dividing each session into its segment components.

For each of the prediction + granularity combinations, we make an 80/20 train-test split. In order to evaluate across multiple trials, we randomly downsample 90% of the total training data, maintaining balanced class sizes ($\min(|S_{\mathrm{H}}|, |S_{\mathrm{G}}|)$ for binary and $\min(|S_{\mathrm{H}}|, |S_{\mathrm{G}}|, |S_{\mathrm{E}}|)$ for ternary), and train each model. There are an average of 1,700 session-level and 9,600 segment-level training samples for binary tasks, and 2,500 session-level, 10,800 segment-level samples for ternary tasks, respectively. The evaluation sets are 593 for `human` vs. `GPT` and 909 for `human` vs. `GPT` vs. `edit`.

4.2. Results

We present the F1-scores for models trained for the AI identification task in Figure 1. We report macro F1 using the unweighted means of precision and recall across classes, as implemented in the scikit-learn library.

In the binary task, classifiers discriminate human-written from AI-generated text with F1-scores at or above 0.8 at the session level. Classification on single segments is harder for all classifiers, consistent with prior work (Chakraborty et al., 2023) where accuracy of neural AI detection models increases as the amount of text to be classified increases. Interestingly, the best performing classifier is the logistic regression model based on LUAR features.

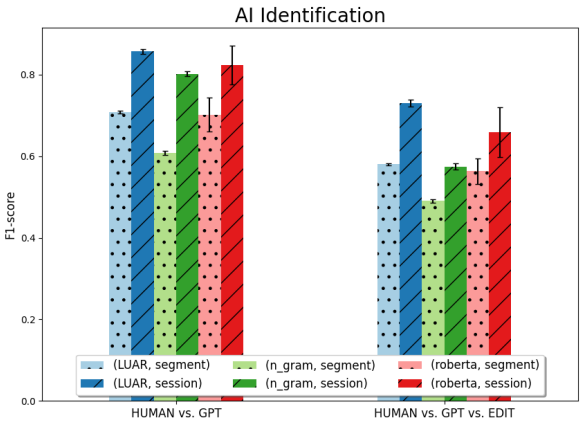

Figure 1: F1-scores for LUAR trained on `human` or `human+GPT+edit` data and evaluated on all four text combination types, averaged over 20 trials.

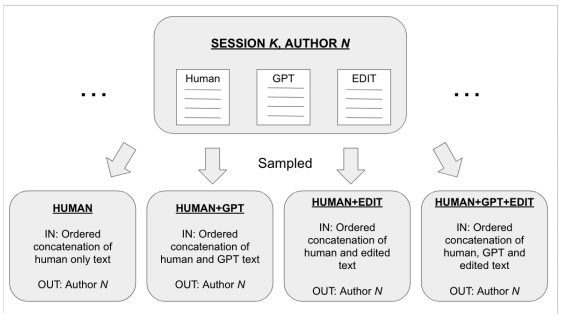

Figure 2: Sketch of method for generating synthetic data for authorship detection.

It outperforms the n-gram-based logistic regression model, possibly because the rich LUAR representations trained on large amounts of Reddit data encode much richer information about text style than the n-gram statistics derived from the training set alone. More surprisingly, the LUAR-based classifier outperforms the RoBERTa-based model which is a standard approach for AI-text detection outside of the co-writing setting (Solaiman et al., 2019).The large standard deviation for the RoBERTa model suggests that fine-tuning such a large model in the low-resource co-writing setting is not as effective as when using large amounts of text independently written by humans and generated by AI. Here, the rich authorship representations from the LUAR models prove more effective and generalize to the task of discriminating human-written from AI-generated text, even though there was presumably no AI-generated text in the LUAR training data.

Moving to the harder ternary task, we find that all classifiers perform worse compared to the binary task, as expected. The LUAR-based model is the most robust of the three, and remains the top performing model.

Overall, these results show that even in the co-writing setting the distributions of `GPT` and `edit` text differ enough from `human` to be discriminated. Interestingly, text representations based on authorship embeddings of human authors prove most effective at this task, even though they were not trained on AI-generated text.

5. Authorship Verification

We turn to the task of authorship verification, a binary classification task which consists of determining whether two sessions are written by the same human author or not. We compare authorship verification classifiers based on sessions comprised of four text combinations (1) `human`, (2) `human` and `GPT`, (3) `human` and `edit` and (4) `human`, `GPT` and `edit`.

5.1. Experiment Setup

Figure 2 diagrams the methodology for creating our synthetic data. Since we want to maintain a consistent collection of sessions across all experiments, we select CoAuthor sessions that contain all three categories of text, and authors that have participated in at least 10 sessions. We start with the collection of sessions S, each containing T_{H}, T_{G} and T_{E} text segments of `human`, `GPT` and `edit` types, respectively. In an attempt to avoid bias from session length and imbalanced subsets of text segments, we use 6 segments as an upper bound on session length. Let S_k be a session and consider $T_{\mathrm{H}_k}, T_{\mathrm{G}_k}, T_{\mathrm{E}_k} \in S_k$, the collection of corresponding text segments. We generate `human` session text by sampling $\min(6, |T_{\mathrm{H}}|)$ segments from T_{H} and concatenating them in the corresponding order in the session. We similarly concatenate sampled text segments by sampling $\min(3, |T_*|)$, $\min(3, |T_*|)$ and $\min(2, |T_*|)$, from the corresponding segment collections when generating `human+edit`, `human+GPT` and `human+edit+GPT` session texts, respectively.

We partition the resulting data into train and test sets according to prompt, so that there is no overlap in training and test prompts. This results in 646 samples for train and 203 for test containing 26 distinct authors. Table 3 summarizes data statistics.

Within this train-test split, we pair sessions written by the same and different authors to create samples for the verification task. This results in a training size and test size of 31,323 and 1,276 instances, respectively, for each text combination category.

By selecting segments of the appropriate authorship type within a session, we train classifiers for authorship verification in each of the three settings:

	Train	Test
No. session	646	203
Avg. no. sess/auth	24.9	7.8
Avg. no. seg/sess	5.7	5.2
Avg. tok len (H)	95.7	98.7
Avg. tok len (HG)	86.8	88.4
Avg. tok len (HE)	102.3	102.2
Avg. tok len (HGE)	101.6	102.0

Table 3: Statistics of co-writing sessions used to train and test authorship analysis classifiers.

- (human, human) input text pairs
- (human + GPT, human + GPT) input text pairs
- (human + GPT + edit, human + GPT + edit) input text pairs

Models are then evaluated on text pairs across the same three settings.

5.2. Results

We compare the four classifier types (Section 3.2) each trained on the three views of the data (human, human+GPT, and human+GPT+edit) on the corresponding versions of the test sets, resulting in 12 models.

Figure 3 (top figure) shows the impact of introducing both types of AI-generated text by comparing the authorship verification performance of these 12 models on human test samples vs. human+GPT+edit test samples. As expected, classifiers based on the LUAR embeddings outperform all other classifiers when evaluating on human written text only (Figure 3 top left), even when they are trained on noisier samples that also contain AI-generated text, with the best model achieving an F1-score of 0.73. The character n-gram model from the PAN shared task is a close second. When evaluating on test samples where each session contains a mix of human+GPT+edit segments (Figure 3 top right), the n-gram based models are more robust to this shift than the LUAR and RoBERTa models. Interestingly, the character n-gram model (cng_dist) is the best performing model in these settings, and its performance remains stable no matter the version of the data it is trained on (human, human+GPT, or human+GPT+edit).

Figure 3 (bottom figure) shows the impact of introducing the edited samples in the data by comparing the same 12 models on human+GPT vs. human+GPT+edit test samples. The character n-gram model (cng_dist) remains the best performing and continues to exhibit stable performance in these new settings. The LUAR-based models are

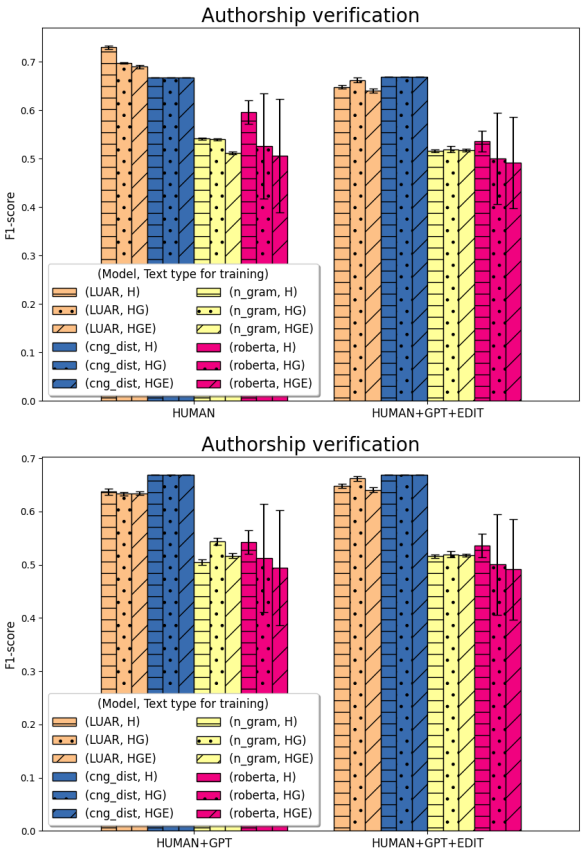

Figure 3: F1-scores for authorship verification models trained and evaluated on (top) human or human+GPT+edit or (bottom) human+GPT or human+GPT+edit data. The error bars are generated from the standard deviation over 5 trials.

impacted by the nature of the AI-generated text in test samples: they perform slightly better when test samples include edit segments (Figure 3 bottom right) in addition to GPT segments (Figure 3 bottom left), indicating that the LUAR embeddings help capture some useful authorship style signal even in text that is edited from LLM output.

Overall, these results show that authorship verification can be achieved with F1-scores reaching 0.67 by the character n-gram classifier, which is more robust to the introduction of AI-generated text and outperforms more complex models based on authorship embeddings and RoBERTa. Perhaps surprisingly, training these models on a mix of human-written and AI-generated text has only limited benefits compared to training on human text only, even when evaluating under the corresponding condition.

5.3. Analysis

In this section, we explore potential reasons behind verification model behavior through feature and error analysis.

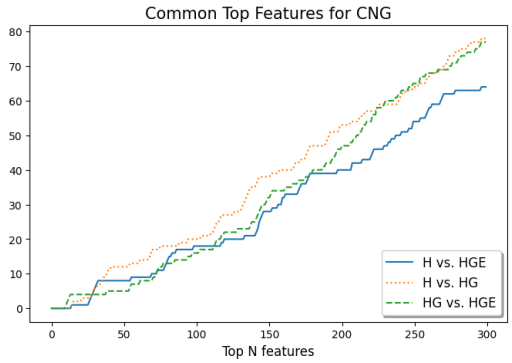

Figure 4: Intersection size of common top N features of the CNG vectorizers trained on `human`, `human+GPT` and `human+GPT+edit` data.

Salient character n-grams Given the stable behavior of the character n-gram model across settings, we investigate whether it relies on the same or different salient features when trained on each version of the data. We plot in Figure 4 the intersection size of the top N character n-gram features across pairwise comparisons of `human`, `human+GPT` and `human+GPT+edit` scenarios. For lower values of N, `human` vs. `human+GPT+edit` and `human+GPT` vs. `human+GPT+edit` have a similar amount of common features. However, as N increases `human+GPT` vs. `human+GPT+edit` and `human+GPT` vs. `human+GPT+edit` tend to be more similar. Although CNG's performance remains close across variants, even at $N = 300$ at most $\frac{1}{4}$ of top features are pairwise common.

We show the common and disjoint sets of features for `human` vs. `human+GPT+edit` in Table 4. The model trained on AI text (second row) relies more on n-grams that contain punctuation and spaces (e.g., "1 da") and expected to be used in conversational registers (e.g., "ok i", "mmmm") than the model trained on human text only (first row) which contains n-grams we would expect to find in longer and rarer words (e.g., "otyp").

Impact of Prompts To understand some of the factors that impact verification performance, we compare the number of errors made by the LUAR verification model based on whether the two texts compared were written in response to the same or different prompts (Figure 5 top). When the prompts are different, introducing AI-generated text in the test samples ((H,HGE) and (HGE,HGE) settings) leads to a bigger increase in errors than when prompts are the same, illustrating that authorship verification is harder when the authorship style and the topic shift simultaneously.

Error types When categorizing samples based on their correct label (Figure 5 bottom), we find that

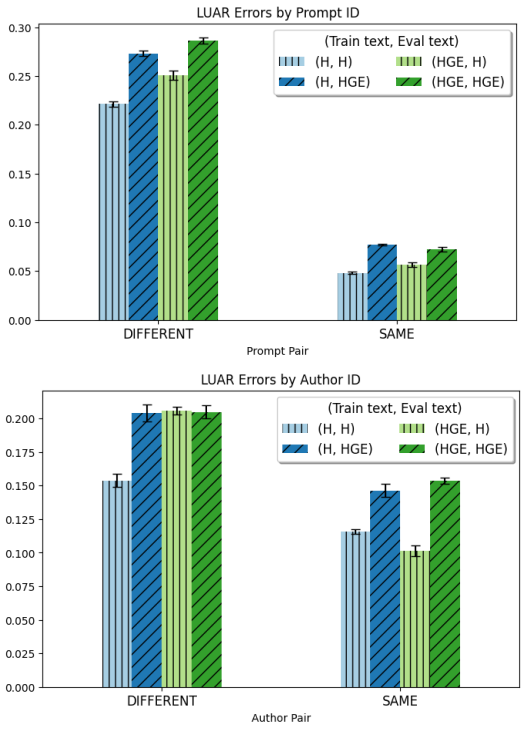

Figure 5: Average percentage of errors of LUAR for the verification task categorized by prompt (top) and author (bottom) ID. The error bars are generated from the standard deviation over 5 trials.

the LUAR model makes more errors by predicting that texts written by different authors are written by the same person (false positive) than failing to detect that two texts are written by the same author (false negative). Using a model trained on human text only to verify authorship on mixed human-AI text leads to a bigger increase in false positives than false negatives, confirming that the injection of AI-generated text makes sessions written by different authors more similar overall. Conversely, when training on mixed human-AI data, evaluating on human-only leaves the number of false positives roughly constant, but decreases the false negatives, indicating that potential inconsistencies in human and AI style within a session hurt prediction accuracy.

6. Authorship Attribution

Finally, we evaluate the impact of AI-generated text in the authorship attribution task, which addresses authorship analysis in a narrower setting than the verification task. In authorship attribution, we assume that training samples for a specific set of authors are available, and that we are only interested in detecting authorship within this closed set. We frame this task as a K-ary classification task on a closed set of $K = 3, 4, 5, 7, 11$ authors.

human	{reot, mpon, rtal, pads, pad, mort, zens, ampo, otyp, mag, tte , ucts, tamp, luc, a bo, eoty, pesh, tam, apes, mmor, immo, sola, hell, olat}
human+GPT+edit	{team, olf., cult, we c, bla, pps , ook., he ",do n, eure, wome, 1 da, d ra, soe, ourc, itic, nged, mmmm, sour, le b, isk , urce, izen, ok i}
Common	{uret, isol, ypes, soeu, oeur, iso}

Table 4: The set of disjoint and common top n-gram CNG features trained on human or human+GPT+edit for $N = 30$. n-grams are not in order of importance.

6.1. Experiment Set-Up

We train models on each category of text combination types (human, human+GPT, human+edit and human+GPT+edit). In order to prevent bias in testing on a specific set of authors, we run multiple trials (t=20) by selecting random subsets of *K* authors within the dataset used for authorship verification. We only use authors that have at least 9 sessions in the test set. All four training variants are evaluated against the four text combination types in the test set.

6.2. Results

We plot the authorship attribution F1-score of the LUAR-based classifier as a function of the number of authors in Figure 6, comparing the combination of two training conditions (human or human+GPT+edit) and four evaluation conditions (human, human+edit, human+GPT and human+GPT+edit). We do not plot the n-gram and RoBERTa classifiers as they underperform the LUAR based classifier by a large margin with F1-scores below 0.5. Note that the character n-gram model is a verification model and is therefore not applicable in the K-ary attribution setting.

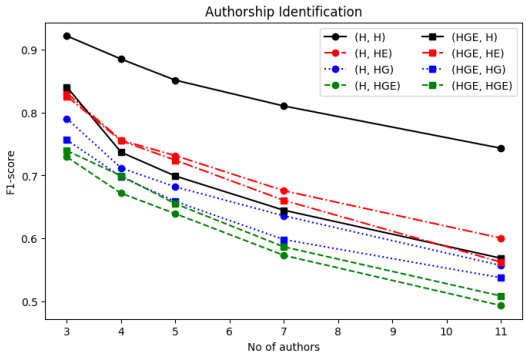

Figure 6: Average F1-scores for LUAR trained on human or human+GPT+edit data and evaluated on all four text combination types.

The human only upper bound – black line (H,H) – shows that the attribution task is increasingly harder when the number of authors increase, with an initial F1-score above 0.9 for only 3 authors down to slightly below 0.8 with 11 authors. All other combinations of train and test configurations degrade attribution performance by 10 F1 points or more. The benefits of training on the same conditions as the test condition are limited, except when testing on human only data.

Overall, these results show that attribution is a hard task with the injection of AI text within a closed set of authors given the data available within CoAuthor. However, when working with a relatively small and closed set of authors, it might be possible to collect additional samples of writing. While it remains to be seen how attribution performance would evolve with the availability of more data, our current results suggest that additional samples of text independently written by human authors might be sufficient to improve accuracy, which might make data collection easier.

7. Conclusion

Using writing sessions from the CoAuthor dataset of human-AI collaborative writing, we conducted a series of experiments to evaluate the impact of injecting AI-generated text on authorship analysis tasks. We found that even when human-written and AI-generated texts are drawn from co-writing sessions, it is possible to distinguish human-written vs. AI-generated text reasonably well (0.8+ F1-score). Classifiers based on the LUAR authorship style embeddings outperform classifiers that are typically used for AI text detection in settings where human and AI text are generated independently. The authorship embeddings proved useful across tasks, including authorship verification – the task of determining whether two texts are written by the same authors, which is more in line with their pre-training objective – and authorship attribution – the classification task for picking an author among a small closed set of candidates, where the LUAR-based classifier outperforms all other approaches. This is notable because the LUAR embeddings are not trained on AI-generated text, and suggests that the underlying representation encodes a rich diversity

of styles which is sufficient to capture the style of AI-generated text.

However, for the authorship verification task, the simpler character n-gram models were more robust to the injection of AI-generated text, outperforming all other classifiers when testing on mixed human-written and AI-generated sessions. Furthermore, in both verification and attribution settings, the benefits of training authorship analysis models on mixed human-written and AI-generated data were limited, indicating that writing samples of existing independently written text might be useful and sufficient when building models to analyse co-authored text.

As any empirical study, our work comes with the limitations associated with the assumptions made in our dataset. Our view of human-AI collaborative writing is thus limited to the affordances provided by the CoAuthor interface, and it remains to be seen how other types of AI writing assistance, such as editing suggestions or iterative revisions, would impact authorship analysis. More data collection would also be needed to study how interactions with GPT-3 might impact the style of authors even when they write from scratch (e.g., entrainment) and whether GPT-generated text differs substantially in style within the context of writing sessions conducted by different participants.

Nevertheless, working with the CoAuthor dataset made it possible to construct controlled experiments comparing different views of sessions, and shows that authorship analysis in those settings remains possible with rich models of authorship style even if they are not trained on AI-generated text. We hope these findings will encourage future work on collecting data and analyzing authorship in a broader range of settings, as the boundary between human-written and AI-generated text continues to blur in years to come.

8. Acknowledgements

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract 2022-22072200006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

9. Bibliographical References

Irshad Abbasi, Abdul Rehman Javed, Farkhund Iqbal, Zunera Jalil, Thippa Gadekallu, and Natalia Kryvinska. 2022. Authorship identification using ensemble learning. *Scientific Reports*, 12.

Janet Ainsworth and Patrick Juola. 2019. Who wrote this?: Modern forensic authorship analysis as a model for valid forensic science. *Washington University Law Review*.

Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Janek Bevendorff, Berta Chulvi, Elisabetta Fersini, Annina Heini, Mike Kestemont, Krzysztof Kredens, Maximilian Mayerl, Reyner Ortega-Bueno, Piotr Pundefinedzik, Martin Potthast, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, Benno Stein, Matti Wiegmann, Magdalena Wolska, and Eva Zangerle. 2022. Overview of pan 2022: Authorship verification, profiling irony and stereotype spreaders, style change detection, and trigger detection: Extended abstract. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II*, page 331–338, Berlin, Heidelberg. Springer-Verlag.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection.

Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *CoRR*, abs/1612.02695.

Hai Dang, Karim Benharrak, Florian Lehmann, and Daniel Buschek. 2022. Beyond text generation:

Supporting writers with continuous automatic text summaries. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA. Association for Computing Machinery.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2022. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text.

Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. BertAA : BERT fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Jiatao Gu, Kyunghyun Cho, and Victor O.K. Li. 2017. Trainable greedy decoding for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1978, Copenhagen, Denmark. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Ting-Hao 'Kenneth' Huang, Vipul Raheja, Dongyeop Kang, John Joon Young Chung, Daniel Gissin, Mina Lee, and Katy Ilonka Gero, editors. 2022. *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*. Association for Computational Linguistics, Dublin, Ireland.

Maurice Jakesch, Advait Bhat, Daniel Buschek, Lior Zalmanson, and Mor Naaman. 2023. Co-writing with opinionated language models affects users' views. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Sylvio Barbon Junior, Rodrigo Augusto Igawa, and Bruno Bogaz Zarpelão. 2017. Authorship verification applied to detection of compromised accounts on online social networks. *Multimedia Tools and Applications*, 76:3213–3233.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense.

Philippe Laban, Jesse Vig, Marti A. Hearst, Caiming Xiong, and Chien-Sheng Wu. 2023. Beyond the chat: Executable and verifiable text-editing with llms.

Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. *CoRR*, abs/2201.06796.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected?

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219, Copenhagen, Denmark. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Efstathios Stamatatos, Mike Kestemont, Krzysztof Kredens, Piotr Pezik, Annina Heini, Janek Bevendorff, Benno Stein, and Martin Potthast. 2022. Overview of the Authorship Verification Task at PAN 2022. In *CLEF 2022 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Efstathios Stamatatos and Moshe Koppel. 2011. Plagiarism and authorship analysis: introduction to the special issue. *Language Resources and Evaluation*, 45(1):1–4.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Min Yang and Kam-Pui Chow. 2014. Authorship attribution for forensic investigation with thousands of authors. In *ICT Systems Security and Privacy Protection*, pages 339–350, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story writing with large language models. In *27th International Conference on Intelligent User Interfaces*, IUI '22, page 841–852, New York, NY, USA. Association for Computing Machinery.