# Overview of the "Voight-Kampff" Generative AI Authorship Verification Task at PAN and ELOQUENT 2024

Janek Bevendorff[1,2], Matti Wiegmann[2], Jussi Karlgren[3], Luise Dürlich[4], Evangelia Gogoulou[4], Aarne Talman[5], Efstathios Stamatatos[6], Martin Potthast[7] and Benno Stein[2]

[1]*Leipzig University*
[2]*Bauhaus-Universität Weimar*
[3]*Silo AI, Helsinki*
[4]*RISE Research Institutes of Sweden, Stockholm*
[5]*University of Helsinki, Helsinki*
[6]*University of the Aegean*
[7]*University of Kassel, hessian.AI, and ScaDS.AI*
*pan@webis.de* *https://pan.webis.de* *https://eloquent-lab.github.io*

**Abstract**

The "Voight-Kampff" Generative AI Authorship Verification task aims to determine whether a text was generated by an AI or written by a human. As in its fictional inspiration,[1] the Voight-Kampff task structures AI detection as a builder-breaker challenge: The *builders*, participants in the PAN lab, submit software to detect AI-written text and the *breakers*, participants in the ELOQUENT lab, submit AI-written text with the goal of fooling the builders. We formulate the task in a way that is reminiscent of a traditional authorship verification problem, where given a pair of texts, their human or machine authorship is to be inferred. For this first task installment, we further restrict the problem so that each pair is guaranteed to contain one human and one machine text. Hence the task description reads: *Given two texts, one authored by a human, one by a machine: pick out the human.*

In total, we evaluated 43 detection systems (30 participant submissions and 13 baselines), ranging from linear classifiers to perplexity-based zero-shot systems. We tested them on 70 individual test set variants organized in 14 base collections, each designed on different constraints such as short texts, Unicode obfuscations, or language switching. The top systems achieve very high scores, proving themselves not perfect but sufficiently robust across a wide range of specialized testing regimes.

Code used for creating the datasets and evaluating the systems, baselines, and data are available on GitHub.[2]

## 1. Introduction

Generative AI is undoubtedly a disruptive technology in the information ecosystem. In particular, large language models (LLMs) have many desirable applications in writing assistance and information access. Although often welcome, unlimited text generation raises concerns in many areas of creation; examples include education and assessment, academic articles and reviews, synthetic misinformation and disinformation, and social bots that influence public discourse. These troubling applications undermine trust in (written) information. Recognizing the fingerprint of AI-generated text thus becomes a promising element for a healthy future information ecosystem, which will become increasingly sophisticated as the fluency and naturalness of the generated text increases.

The Voight-Kampff task investigates the feasibility of identifying whether text is written by a human author or generated by a language model. We recognize that the detection of AI-generated text is closely related to the identification of human authorship where, prospectively and with increasing fidelity and diversification of models, each AI model can be considered an author that exhibits particular and identifiable characteristics [1]. In this task, we adapt this idea and formulate AI detection as an authorship problem. This allows us not only to draw upon experience from previous LLM detection shared tasks [2, 3, 4, 5], but also to adapt decades of theoretical and engineering work on author identification, including past work at PAN [6, 7, 8, 9, 10, 11].

[1] In the movie "Blade Runner", the eponymous officers use the Voight-Kampff machine to test whether a subject is a replicant.
[2] Code and data: https://github.com/pan-webis-de/pan24-generative-ai-authorship-verification

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

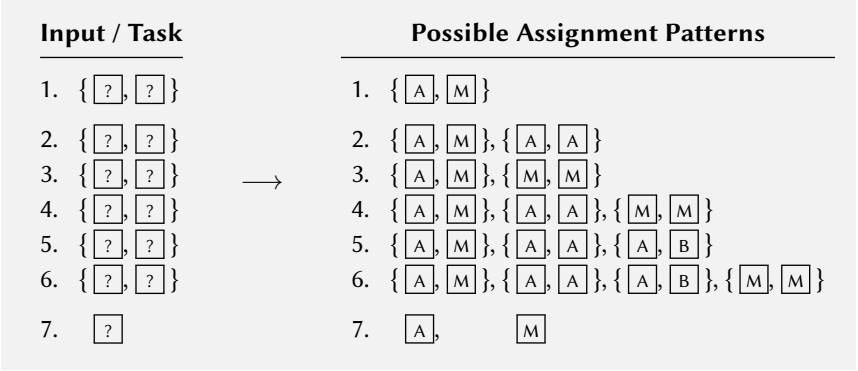


**Figure 1:** Hierarchy of authorship verification problems from *easiest* (1) to *hardest* (7), involving LLM-generated text. Ignoring mixed human and machine authorship, the difficulty arises from the pairing constraints imposed by the possible assignment patterns. [M] denotes LLM-generated text, while [A] and [B] denote human-authored text (same letter meaning same human author).

Identifying a single disputed document as AI-generated without reference is an open problem and the most difficult formulation of the AI detection task. Although the literature suggests limited success in solving this problem with the current generation of LLMs, it is questionable whether this will remain the case as the technology improves. Aside from mixed human-machine authorship, we have broken down the relevant formulations of the problem with increasing levels of difficulty to gain a more fundamental understanding of the task at hand and the feasibility of potential solutions. Figure 1 visualizes the cascade of all problem variants from the easiest (Task 1) to the most difficult (Task 7). In the easiest case, two documents of unknown authorship are given, but we guarantee that exactly one of them was created by a human [A] and the other by a machine [M]. This constraint is relaxed in the following variants, where e.g., both texts can be generated by a machine, {[M], [M]}. In the hardest case, a single text is given, which could be either [A] or [M].

The Voight-Kampff task follows the *easiest* formulation of the task to establish a feasibility baseline. The task description is: *Given two texts, one authored by a human, one by a machine: pick out the human.*

Our evaluation campaign was organized in a builder-breaker setup in collaboration between the ELOQUENT and PAN labs at CLEF. In this setup, both systems and evaluation resources are built by the participants and compete in an adversarial setting. The *builders* at PAN create analysis tools, in this case for classifying texts as human-authored vs. machine generated. The *breakers* at ELOQUENT provide the data points to break the analysis tools, in this case to fool the classifiers into believing that a machine-generated text was authored by a human. This adversarial design is intended to focus participants' efforts by providing increasingly challenging baselines for both classifiers and generative language models.

In total, our evaluation campaign includes 14 breaker dataset collections consisting in total of 70 individual conditions (or variants), eight provided by PAN as baselines and six submitted by ELOQUENT participants. We evaluate 43 builder systems, 13 of which were provided by PAN as baselines and 30 that were submitted by PAN participants. We find that the strongest systems are highly effective and robust across dataset variants in separating machine-generated and human-authored text, at least given the latest generation of large language models. The most difficult evaluation settings across all submitted systems are unexpected languages and very short texts, closely followed by Unicode obfuscations.

## 2. Detecting AI-generated Text

In total, 30 valid system were submitted to the builder task in addition to the baseline systems we provided. Of these, 28 teams submitted descriptions of their systems in the form of notebook papers. Table 4 shows the final system ranking and Table 1 shows an overview of the systems.

## 2.1. Baselines

We provided implementations of six baseline systems to compare submitted systems against four state-of-the-art zero-shot LLM detection baselines and two adapted authorship verification baselines.

The zero-shot LLM detection baselines are: (1) Binoculars [12], (2) DetectLLM [13] (both NPR and LRR scoring mode), (3) DetectGPT [14], and (4) Fast-DetectGPT [15]. All three were provided in two variants using either Falcon 7B [16] or Mistral 7B [17] to estimate text perplexities. The required text perturbations for DetectGPT and DetectLLM-NPR were generated with T5-3B [18].

The two authorship verification baselines were adapted to the LLM detection task by splitting each text in half and comparing the two halves against each other under the assumption that LLM texts are stylistically more self-similar than human texts. The baselines provided are a compression model (PPMd Compression-based Cosine) [19, 20] and short-text authorship unmasking [21, 22].

As an additional seventh baseline, we measured and compared the text lengths in characters. This baseline serves as both a quasi-random baseline and as a data sanity check.

## 2.2. Builder Submissions to PAN

While our baseline systems reproduce established methods in either authorship verification or intrinsic, zero-shot LLM detection, the participant systems cover a broad range of approaches. Table 1 shows an overview of the essential elements across all systems. The most frequent approach is to train or fine-tune a (neural) classifier with term-based features, in most cases (BERT) embeddings. Many systems either apply some training regime modification (such as PU loss or R-Drop), use ensembles, and / or expand the given training data with other LLM detection datasets. Some systems use engineered features like perplexity (PPL), term frequencies (TF), stylometric features (text length, properties of token distributions), or a combination of these.

The systems use a broad range of classification models (SVM, XGBoost, LSTM, CNN) but in most cases in combination with BERT. Most systems only classify if one given document is machine generated and decide which of the two input documents is human-written by comparing the predicted probability, similar to how the provided baselines do it. Some participants, however, also trained models to directly discriminate between the pairings as in the traditional author verification setting. In some cases, participants also utilized LLMs for the detection task, often using Low-Rank adaptation (LoRA).

### 2.2.1. Term-based Systems

Huang et al. [24], the second-ranked system (0.921), derives a method named Tri-Sentence Analysis (TSA) from the multi-scale positive-unlabeled (MPU [51]) LLM-detector. TSA dissects the documents into small (3-sentence) chunks and trains a BERT classifier for binary AI detection on the chunks, where each chunk inherits the class label of the document. The chunk scores are then averaged to to estimated the prediction on the original documents.

Lorenz et al. [25], the third-ranked system (0.886), uses feature-based, supervised classification: Naive Bayes, Logistic Regression, and linear SVM based on the top 1,000 TF-IDF term count features.

Guo et al. [26] (0.884) use a supervised hybrid method that utilizes various types of BERT embeddings and Google Books word frequency features embedded with an LSTM.

Several other mostly well-performing models fine-tuned a BERT (variant) with simple modifications: Lin et al. [27] (0.851) with R-Drop [52] regularization, Yadagiri et al. [30] (0.806) with augmentation through linguistic features (vocabulary count, a word-density measure, and POS counts), Lv et al. [31] (0.804, DeBERTa) with Reptile meta-learning [53], Cao et al. [33] (0.778) with dataset augmentation, Huang et al. [35] (0.735) with R-Drop in combination and additional training data from Kaggle [54] for oversampling the human examples, Petropoulos and Petropoulos [41] (0.641) with a Bi-LSTM between the RoBERTa and the linear classification layer, Wu et al. [42] (0.608) with an extra transformer-layer between the BERT and the linear classification layer, Zhu and Kong [44] (0.555) by using DeBERTa instead of BERT, Sun et al. [45] (0.531) with a 2D-CNN-layer between a BERT and a linear layer, and Huang et al. [48] (0.480) with a siamese DeBERTa with contrastive learning and domain adaptation.

**Table 1**
Systems Overview. Shown is an overview of the kind of features used (contextualized embeddings, LLM-based text perplexity (PPL), term frequency vectors (TF), or other stylometric / linguistic features), whether the system is an ensemble, whether the training data was augmented, and whether the classifier was zero-shot or a supervised learned fit to training data.

| | Team | Features | | | | Ensemble | Data Aug. | Zero-shot |
|---|---|---|---|---|---|---|---|---|
| | | Embed. | PPL | TF | Style | | | |
| 1 | Tavan [23] | x | x | | | x | | (x) |
| 2 | J. Huang [24] | x | | | | | | |
| 3 | Lorenz [25] | | | x | | | | |
| 4 | M. Guo [26] | | x | | x | | | |
| 5 | Zi. Lin [27] | x | | | | | | |
| 6 | Abburi [28] | x | x | | | | | |
| 7 | Miralles [29] | | x | | x | | | |
| 8 | Yadagiri [30] | x | | | x | | | |
| 9 | Lv [31] | x | | | | | | |
| 10 | Gritsai [32] | x | | | | x | | |
| 11 | Cao [33] | x | | | | | x | |
| 12 | L. Guo [34] | x | | | x | | | |
| | *Binoculars (Falcon-7B)* [12] | | x | | | | | x |
| 13 | B. Huang [35] | | x | | | | x | |
| 14 | Valdez-Valenzuela [36] | x | | | x | | x | |
| 15 | Ye [37] | x | | | | | x | |
| 16 | Chen [38] | x | x | | | x | x | |
| 17 | W. Huang [39] | | x | | | | | |
| 18 | Qin [40] | | x | | | x | x | |
| | *Binoculars (Mistral-7B)* [12] | | x | | | | | x |
| | *DetectLLM LRR (Mistral-7B)* [13] | | x | | | | | x |
| 19 | Petropoulos [41] | x | | | | | | |
| | *Fast-DetectGPT (Mistral-7B)* [15] | | x | | | | | x |
| 20 | Z. Wu [42] | x | | | | | | |
| | *Text Length* | | | | x | | | x |
| 21 | *gra*† | | | | | | | |
| 22 | Zh. Lin [43] | x | | | | | | |
| 23 | Zhu [44] | x | | | | | | |
| | *PPMd Compression-based Cosine* [19, 20] | | | | | | | x |
| 24 | Sun [45] | x | | | | | | |
| | *DetectLLM NPR (Mistral-7B)* [13] | | x | | | | | x |
| 25 | Lei [46] | x | | | | | | |
| | *Fast-DetectGPT (Falcon-7B)* [15] | | x | | | | | x |
| 26 | Liu [47] | | x | | | | | x |
| 27 | *e-comm-tech*† | | | | | | | |
| | *DetectGPT (Mistral-7B)* [14] | | x | | | | | x |
| 28 | K. Huang [48] | x | | | | | | |
| | *DetectLLM NPR (Falcon-7B)* [13] | | x | | | | | x |
| | *Authorship Unmasking* [21, 22] | | | | | | | x |
| 29 | Sheykhlan [49] | | x | | | x | | |
| | *DetectLLM LRR (Falcon-7B)* [13] | | x | | | | | x |
| 30 | G. Wu [50] | x | | | | | | |
| | *DetectGPT (Falcon-7B)* [14] | | x | | | | | x |
| | Sum of participant systems | 20 | 11 | 1 | 5 | 5 | 6 | 0 |

† No notebook submitted.

Two systems build on contextualized word embeddings from BERT but use a more involved model: Guo et al. [34] (0.763) uses a Bi-LSTM followed by a transformer layer for classification. As input for the LSTM, the authors use BERT embeddings concatenated with seven stylometric and linguistic features (lexical diversity, average sentence length, average word length, the number of grammatical errors, sentiment tendency, repetition rate, and stop word ratio). Valdez-Valenzuela and Gómez-Adorno [36] (0.727) use a mixture of co-occurrence graph features embedded with a GNN, stylometric features, and

BERT document embeddings and augment the training data with additional human texts.

Two systems use ensembles based on multiple fine-tuned BERT models: Qin et al. [40] (0.680) use a voting ensemble of a basic BERT and a BERT with R-Drop regularization, trained on additional data from Kaggle. Sheykhlan et al. [49] (0.460) use a voting ensemble with BERT, RoBERTa, and Electra.

Finally, five systems use generative LLM as base for a classifier: Gritsai et al. [32] (0.796) use an ensemble of multiple Mistral models, each fine-tuned via QLoRA on texts generated by different types of LLMs. Ye et al. [37] (0.722) fine-tune a T5 model with language modeling head to predict the tokens "positive" (machine text) or "negative" (human text) for a given document. If neither of these tokens is the most likely, the system outputs "undecided". Lin et al. [43] (0.565) also fine-tune a T5 to predict "positive" or "negative" after a new, special token and assign the probability of whichever token is more likely. Lei et al. [46] (0.504) fine-tune a ChatGLM model for authorship attribution, i.e. the model learns to predict tokens that indicate either "Human" any one of the particular LLMs. The predicted classes are then transformed back into a binary AI detection score. Wu and Guan [50] (0.450) use a language model pre-trained for NLI under the assumption that LLM-generated texts show weaker coherence and textual entailment between sentences.

### 2.2.2. Perplexity-based Systems

Miralles et al. [29] (0.806) use an XGBoost classifier with features (mean, stddev, etc.) of the distributions of next-token probabilities across five current LLMs, in combination with established, stylometric features.

Huang and Grieve [39] (0.683) use the perplexity of "authorial language models" as features for an SVM classifier. As authorial language, a GPT-2 is fine-tuned for each known LLM and one for all humans. For both disputed text, the perplexity of all authorial LMs is measured and provided as feature vector for the SVM. The SVM is then trained for verification.

Liu and Kong [47] (0.497) use the perplexity of a GPT-2 model as discriminator under the assumption that the text with the lower GPT-2 perplexity score is AI-generated. The perplexity is calculated as a sum on a sliding 1,024-token window.

### 2.2.3. Systems Using Terms and Perplexity

Tavan and Najafi [23], the first-ranked system (0.924), uses an ensemble of two LLMs (Mistral and Llama2) and the Binoculars baseline. The LLMs (with classification head) were fine-tuned on the training data via LoRA. The final score is the average decision across all three models. Abburi et al. [28] (0.843) use a combination of a RoBERTa-based AI detector, token-level probability features from multiple GPT-2 variants and E5 document embeddings to classify AI texts. Finally, Chen and Kong [38] (0.694) use a voting ensemble of three models: 2× BERT and a GPT-2. The BERT models predict which of two given texts is written by a human, the first or the second. Trained on different splits of the training data (with inverted positions and labels). Both models are identical, but the input pairs are flipped to counteract the truncation of long documents. The GPT-2 model is used to calculate the perplexity of each text (for ca. 500 character chunks and summed for each text), where text with the higher perplexity is taken as the human written one. The training data is augmented with the Kaggle DAIGT v2 Train [55] dataset.

## 3. Datasets

We used two dataset collections in the evaluation of the "Voight-Kampff" task. The first collection, in accordance with the builder-breaker pattern, collects all datasets submitted by individual ELOQUENT participants. This collection is described in detail in Section 3.1.

As both labs were ran concurrently, we created a second "bootstrap" dataset collection `PAN AI News 2021` in lieu of training data to get participants started. Participants were encouraged to also use additional data from other sources; Part of this bootstrap collection was held back as a test collection.

**Genre and Style**:

The text is an informative piece providing a comprehensive overview of Malaysia's geography, history, government structure, economy, and cultural diversity. Its tone is neutral and factual, aiming to educate the reader about various aspects of the country.

**Content**:

- Malaysia is a federal constitutional monarchy in Southeast Asia, comprising thirteen states and three federal territories.
- Geographically divided into Peninsular Malaysia and East Malaysia (Malaysian Borneo) by the South China Sea.
- Shares borders with Thailand, Singapore, Vietnam, Indonesia, Brunei, and maritime borders with the Philippines.
- Capital city: Kuala Lumpur; federal government seat: Putrajaya.
- Multi-ethnic and multi-cultural country with Islam as the state religion, but freedom of religion is protected.
- Boasts a strong economy, historically driven by natural resources but expanding into sectors like science, tourism, commerce, and medical tourism.

**Figure 2:** A sample summary for the Voight-Kampff breaker task.

The `PAN AI News 2021` dataset consists of news articles written by humans or LLMs and its creation is described in Section 3.2.

## 3.1. Breaker Submissions to ELOQUENT

ELOQUENT formulated the breaker challenge for this task, with the objective for participants to use models and systems of their choice to fool classifiers into believing their output is authored by a human. The organizers selected 29 human authored texts, five sample items for pre-experiment tuning and testing purposes, and 24 items proper. Each text was of 300 to 600 words length and summaries of each text were generated by the organizers using OpenAI's ChatGPT service with the prompt:

```
Summarize the following text in five to six short bullet points and give an overall
description of the genre and tone of the text.
```

Those machine-generated summaries were then shared with the participants, so their systems could generate derivative short texts. A sample summary is given in Figure 2 and a list of all test item titles are given in Table 2. We suggested the following prompt but the participants were free to formulate their own prompts as they saw fit.

```
Write a text of about 500 words which covers the following items:
```

The task had 35 registered teams. By the deadline three teams participated, with five experimental conditions submitted. The models used are Poro [56] and Mistral [57] submitted by team Reindeer [58], GPT-SW3 [59], a RAG-enhanced system based on the Command-R model submitted by team "Verbanex" from Universidad Tecnológica de Bolívar, and a GPT 3.5-based baseline produced by the organizers. Poro is a decoder-only model with a parameter count of 34 billion and 54 layers, trained on the LUMI supercomputer with 1 trillion tokens for Finnish, English, and code. In testing, Poro has been found to be reasonably competent in several other languages as well, due to the multilinguality of the Finnish data set. The Mistral model was used as a comparison since it is better instruction trained for conversational data. Poro and Mistral are open source models, freely available for use in experimentation. GPT-SW3 is based on the GPT-3 architecture and trained on the Berzelius supercomputer with 300B tokens for Swedish, Norwegian, Danish, Icelandic, English, and code. GPT-SW3 is available for research purposes. The Command-R series of models, built for RAG with a longer input context than many other models, are tested for quality in several languages including English and are available for research purposes.

**Table 2**
Items for the Voight-Kampff breaker task. Those were given to the participants as clues to generate test texts.

| ID | Title | Source |
|---|---|---|
| 001 | Uralic languages | Encyclopedia Britannica |
| 002 | Taylor and Travis | Washington Post |
| 003 | Relationships the Good and the Messy | Podcast transcript |
| 004 | A Day of Very Low Probability | Fan fiction |
| 005 | How to Cope With Anxiety-Induced Procrastination | Lifehack website |
| 006 | Malaysia | Wikipedia |
| 007 | Alps | Wikipedia |
| 008 | 2008 Summer Olympics | Wikipedia |
| 009 | Peter Higgs | Encyclopedia Britannica |
| 010 | Richard Serra | Encyclopedia Britannica |
| 011 | Johann Eck | Encyclopedia Britannica |
| 012 | 1000 Things Worth Knowing That all who read may know | Gutenberg |
| 013 | Robert Elsmere | Gutenberg |
| 014 | An Inquiry into the Nature and Causes of the Wealth of Nations | Gutenberg |
| 015 | Dyslexia Basics | International Dyslexia Association |
| 016 | Textual stylistic variation: Choices, genres and individuals | Arxiv |
| 017 | The Fëanorieli by Istarnie | Council of Elrond Tolkien appreciation site |
| 018 | Star Moors | Archive of our own fiction site |
| 019 | Spirit of Strife | Archive of our own fiction site |
| 020 | Eggplant Parmesan | Brown Eyed Baker recipe site |
| 021 | Easy Homemade Ramen Bowls | Killing Thyme recipe site |
| 022 | Vegan Tiramisu | Lazy Cat Kitchen recipe site |
| 023 | HEA Warns of Growing Third Level Funds Crisis | Irish Times |
| 024 | New Artwork Celebrating 100 years of Women in Law | UK Supreme Court |
| 025 | ELOQUENT shared tasks for evaluation of generative language model quality | ELOQUENT paper |
| 026 | 3 Baltic Capitals | Travel tips newsletter |
| 027 | A Guide to the Principles of Animal Nutrition | Oregon State University |
| 028 | The Great Days of the Clippers | Gutenberg |
| 029 | The Three Musketeers | Gutenberg |

## 3.2. The "PAN AI News 2021" Dataset Collection

For creating the second dataset collection, we first scraped 1,359 articles of major 2021 U.S. news headlines from Google News, then generated summaries of each article, and finally re-generated news-alike articles from these using nine large language models.

We scraped Google News using the GNews Python library[2] and Newspaper3k[3] to download the articles and extract the plain texts. We chose the year 2021 specifically as it predates the release of GPT-3.5 so that we could be reasonably certain the articles were actually human-authored.

Using the plain texts as input, we instructed GPT-4 Turbo to generate a bulleted summary of each article as shown in Figure 3. To be able to generate convincing and high-quality articles with a high similarity to the human texts, we also extracted (1) the article's type (nine classes), (2) the target audience (three classes), (3) the authors political stance (three classes), (4) the articles dateline, and (5) the names and functions of directly quoted spokespersons, if any. The result was to be returned as JSON with a predefined schema (the output was mostly valid, though some syntax and schema errors had to be corrected later by hand).

Given the so-generated summaries, we prompted several instruction-tuned downstream LLMs to assume the role of a journalist from the respective source medium in writing an article of the given type about the extracted key points. The article should have the same stance, target audience, and start

[2] https://github.com/ranahaani/GNews
[3] https://github.com/codelucas/newspaper

**Summary** You are a news article and press release summarizer. Given an article, you summarize the key points in 10 bullet points.

**Type** You also classify the article type ("breaking news", "press release", "government agency statement", "financial news", "opinion piece", "fact check", "celebrity news", "general reporting", "speech transcript").

**Dateline** Extract the dateline from the beginning of the article if one exists (e.g. "WASHINGTON " or "May 28 (Reuters)").

**Quotes** If spokespersons are cited verbatim, list their names, functions, and titles (if any).

**Audience** Determine the article's target audience ("general public", "professionals", "children").

**Stance** Classify whether the article's stance is "left-leaning", "right-leaning", or "neutral".

**Structure** Answer in structured JSON format (without Markdown formatting) like so:

```
{
    "key_points": ["key point 1", "key point 2", . . . ],
    "spokespersons": ["person1 (title, function)", . . . ],
    "article_type": "article type",
    "dateline": "dateline",
    "audience": "audience",
    "stance": "stance"
}
```

**Figure 3:** Prompt used for "PAN AI News 2021" to generate article summaries and extract style information to be used in re-generating articles.

with the same dateline. Direct quotations from the originally cited spokespersons were to be included as well, though we did not prescribe what those spokespersons were alleged to have said.

In particular, we used the following LLMs for generating the articles:

1. GPT-3.5 Turbo [60]
2. GPT-4 Turbo [61]
3. Gemini Pro [62] (with temperatures of 0.6 and 0.9)
4. PaLM2 Text-Bison [63]
5. Meta Llama2 7B / 13B / 70B Chat [64]
6. Mistral 7B Instruct v0.2 [17]
7. Mixtral 8x7B Instruct v0.1 [65]
8. BLOOMZ 7B1 [66]
9. Qwen-1.5 72B Chat [67] (8-bit-quantized).

Unless stated otherwise, we used the API default settings for GPT, Gemini, and PaLM. The remaining models were retrieved from Huggingface [68] to run on our own infrastructure. Llama2 13B was used with two different settings for contrastive decoding [69] ($\alpha = 0.1$ and $\alpha = 0.6$). We also planned to include Falcon 7B and 40B Instruct [16], but were unable to get sensible articles out of it.

In addition to the LLMs above, we also used a GPT-2 model fine-tuned on the Open-Instruct dataset [70] and two Alpaca models [71] based on Llama2 7B and 13B as lower-quality baselines (with a shortened prompt to fit the smaller input sizes).

### Text Pre- and Post-Processing

We manually reviewed all of the extracted plain text from the human-written articles that we scraped from Google News and removed any remaining artifacts, such as page footers, navigation fragments, or figure captions.

Similarly, we manually reviewed all LLM responses and thoroughly removed all obvious artifacts that might give away the LLM authorship too easily. We removed typical LLM chat phrases such as *"Sure, I'd*

**Table 3**
Overview of the 65 variants of the "PAN AI News 2021" test set. All variants are seeded from the same 272 human texts from the Google News 2021 holdout set. The "main" and "cross-topic" variants contain all 13 × 272 possible human-LLM pairs minus a few failed generations (e.g., due to moderation guardrails or insufficient output length). Unicode substitutions and short texts are random 600-pair subsets of these. Contrastive decoding, German texts, and the Kaggle prompt variant make use of only one or two LLMs each, resulting in fewer pairs.

| | | **Alpaca** | **BLOOMZ** | **Gemini Pro** *with temp.* | | **Text-Bison** | **GPT** | | | **Llama2** | | | **Mistral** | | **Qwen-1.5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Test Set Variant** | **Pairs** | 7B | 7B1 | 0.6 | 0.9 | 002 | 2-OI | 3.5 | 4 | 7B | 13B | 70B | 7B | 8x7B | 72B |
| Main | 3,411 | x | x | x | x | x | x | x | x | x | | x | x | x | x |
| Unicode substitution (machine) | 600 | | | x | x | | x | x | x | x | | x | | | |
| Unicode substitution (both) | 600 | | | x | x | | x | x | x | x | | x | | | |
| Cross-topic | 3,411 | x | x | x | x | x | x | x | x | x | | x | x | x | x |
| Short text (35 words) | 600 | x | x | x | x | x | x | x | x | x | | x | x | x | x |
| German text (machine) | 543 | | | | | x | | | x | | | | | | |
| Contrastive decoding ($\alpha = 0.1$) | 272 | | | | | | | | | | x | | | | |
| Contrastive decoding ($\alpha = 0.6$) | 272 | | | | | | | | | | x | | | | |
| Kaggle prompt | 542 | | | x | | | | | x | | | | | | |

*be happy to help.", "Sorry, I cannot…", "Here's your article:", "Here are 10 paragraphs:"* or *"In this article, I will…"*, markers and placeholders such as *"[your name]", "[email]"* or *"[end of article]"*, but also more complex structural artifacts. Typical structural artifacts included numbering of individual paragraphs, excessive use of bulleted lists, or newlines after the dateline. A very peculiar artifact many LLMs exhibited was to append a bulleted list of "quotations" from the spokespersons to the end of the article instead of incorporating them into the article itself. Some LLMs also liked giving (very "approximate") word or paragraph counts at the end. With the exception of Llama2, the open-source LLMs tended to produce more such artifacts than the closed-source LLMs.

As a final post-processing step, we truncated the (generally longer) human texts in the bootstrap dataset to approximately the same length as the average LLM text. We did this by fitting a log-normal distribution to the LLM text lengths and truncated the human texts accordingly by drawing from this distribution. To avoid cutting texts in the middle of sentences, we used the drawn text lengths only as a starting point to search for the nearest paragraph ending within a window of at most 200 characters.

### Bootstrap and Main Test Split

We initially published 1,087 of the original 1,359 human-authored articles, together with the re-generated counterparts from 13 of the 16 LLMs. This "bootstrap" (i.e., training) dataset was released so the participants could calibrate their systems. The texts from each LLM were collected in a separate newline-delimited JSON files together with one file for all human texts.

The remaining 272 human-authored articles and their counterparts were kept back for testing. This resulted in 3,984 test cases (pairs of human and generated texts), which together form the **main** (sub-)collection of the test set. The test set is formatted also as newline-delimited JSON, but in a single file. Each line contains one of the original human texts and its corresponding machine generations as a random-order pair. Texts within a pair were cut to the same number of words within a window of 65 words, trying to preserve full sentences if possible.

Notably, all generated text variants from Llama2 13B with contrastive decoding and the higher-temperature variant of Gemini Pro were excluded completely from the "bootstrap" data to test how robust detectors are to unseen models. In contrast, all variants from Alpaca 13B were added only to the bootstrap dataset.

**Test Data Variations for Robustness Evaluation**

To further test the robustness of the submitted systems against certain text modifications, we generated multiple, unknown variants of the original test cases:

1. For **German Text**, we amended the prompt to generate German instead of English texts. This was already part of the **Main Test** set but not of the initially published bootstrap dataset.
2. For **Unicode**, we replaced 15 % of the characters in (a) the machine texts and (b) both the human and machine texts with Unicode lookalike characters.
3. For **Cross-topic**, we shuffled the test case pairings to break the topic coherence.
4. For **Contrastive Decoding**, we used contrastive decoding [72] instead of top-$k$ / top-$p$ sampling;
5. For **Short Text**, we cropped the texts to 35 words; and
6. For **Kaggle Paraphrase**, we used the prompt from a previous Kaggle competition on LLM detection [2] to generate more faithful and direct paraphrases of the original articles:

   ```
   The following is a human-written article.  Now, please rewrite this article in
   your writing style, also optimize sentence structures and correct grammatical
   errors.  You can appropriately add or remove content associated with the
   article, but should keep the general meaning unchanged.  Just return the
   modified article.
   ```

   It is followed by the original text instead of the bullet-point summary.

In total, we created 65 test set variants from the 13 source LLMs and the eight conditions given above. Table 3 shows a systematic overview of all variants.

## 4. Evaluation

At test time, participants were given pairs of human and LLM texts and had to calculate a score between 0 and 1 indicating which text was more likely to be human. Scores below 0.5 indicate that the left text is human, and scores above 0.5 indicate that the right text is human. A score of exactly 0.5 could be given to signal a non-decision. We borrowed this scoring scheme from previous iterations of the PAN authorship verification task.

We evaluate the overall *effectiveness* of the systems on each dataset variant as the arithmetic mean of the following five metrics, which have also been used for evaluation in previous PAN shared tasks on authorship verification (all with comparable 0–1 scales):

- Roc-Auc: The area under the Receiver Operating Characteristic curve.
- Brier: The complement of the Brier score, which is in our case equivalent to the mean squared loss.
- C@1: A modified accuracy score that assigns non-answers (score = 0.5) the average accuracy of the remaining cases.
- $F_1$: The harmonic mean of precision and recall.
- $F_{0.5u}$: A modified $F_{0.5}$ measure (precision-weighted F measure) that treats non-answers (score = 0.5) as false negatives.

All metrics were corrected by discounting half a standard deviation, estimated on each dataset individually, from the system's scores with $n - 1$ degrees of freedom. This penalizes unstable systems with widely varying scores on the individual dataset variants, and promotes systems that are more robust to text obfuscation or other text modifications (even if their mean performance may be slightly worse than that of other systems). We decided to use the macro average across all datasets because, even though the datasets have different numbers of examples, we consider all datasets equally important as performance indicators.

**Table 4**
Final PAN leader board. Systems are ranked by the mean of all evaluation measures across all other metrics on the main dataset discounted by half a standard deviation to correct for spread.

| | Team | ROC-AUC | Brier | C@1 | $F_1$ | $F_{0.5u}$ | Mean |
|---|---|---|---|---|---|---|---|
| 1 | Tavan [23] | **0.961** | **0.928** | 0.912 | 0.884 | **0.932** | **0.924** |
| 2 | J. Huang [24] | 0.931 | 0.926 | **0.928** | **0.905** | 0.913 | 0.921 |
| 3 | Lorenz [25] | 0.925 | 0.869 | 0.882 | 0.875 | 0.869 | 0.886 |
| 4 | M. Guo [26] | 0.889 | 0.875 | 0.887 | 0.884 | 0.884 | 0.884 |
| 5 | Zi. Lin [27] | 0.851 | 0.850 | 0.850 | 0.852 | 0.849 | 0.851 |
| 6 | Abburi [28] | 0.866 | 0.863 | 0.834 | 0.825 | 0.820 | 0.843 |
| 7 | Miralles [29] | 0.831 | 0.825 | 0.795 | 0.788 | 0.782 | 0.806 |
| 8 | Yadagiri [30] | 0.844 | 0.793 | 0.805 | 0.789 | 0.792 | 0.806 |
| 9 | Lv [31] | 0.833 | 0.867 | 0.799 | 0.748 | 0.767 | 0.804 |
| 10 | Gritsai [32] | 0.853 | 0.862 | 0.795 | 0.718 | 0.742 | 0.796 |
| 11 | Cao [33] | 0.777 | 0.777 | 0.777 | 0.780 | 0.777 | 0.778 |
| 12 | L. Guo [34] | 0.799 | 0.788 | 0.740 | 0.740 | 0.741 | 0.763 |
| | *Baseline Binoculars (Falcon 7B)* [12] | 0.751 | 0.780 | 0.734 | 0.720 | 0.720 | 0.741 |
| 13 | B. Huang [35] | 0.756* | 0.782* | 0.726* | 0.706* | 0.703* | 0.735* |
| 14 | Valdez-Valenzuela [36] | 0.741* | 0.760* | 0.718* | 0.711* | 0.695* | 0.727* |
| 15 | Ye [37] | 0.901 | 0.758 | 0.733 | 0.549 | 0.653 | 0.722 |
| 16 | Chen [38] | 0.692 | 0.678 | 0.678 | 0.732 | 0.680 | 0.694 |
| 17 | W. Huang [39] | 0.736 | 0.731 | 0.731 | 0.590 | 0.614 | 0.683 |
| 18 | Qin [40] | 0.689* | 0.730* | 0.672* | 0.652* | 0.652* | 0.680* |
| | *Baseline Binoculars (Mistral 7B)* [12] | 0.676 | 0.711 | 0.663 | 0.654 | 0.648 | 0.671 |
| | *Baseline DetectLLM LRR (Mistral 7B)* [13] | 0.656 | 0.758 | 0.617 | 0.618 | 0.618 | 0.654 |
| 19 | Petropoulos [41] | 0.594 | 0.694 | 0.670 | 0.631 | 0.590 | 0.641 |
| | *Baseline Fast-DetectGPT (Mistral 7B)* [15] | 0.637 | 0.710 | 0.616 | 0.611 | 0.608 | 0.638 |
| 20 | Z. Wu [42] | 0.645 | 0.649 | 0.587 | 0.578 | 0.577 | 0.608 |
| | *Baseline Text Length* | 0.608 | 0.607 | 0.607 | 0.596 | 0.596 | 0.604 |
| 21 | *gra*† | 0.500 | 0.750 | 0.467 | 0.634 | 0.521 | 0.574 |
| 22 | Zh. Lin [43] | 0.593 | 0.598 | 0.598 | 0.458 | 0.565 | 0.565 |
| 23 | Zhu [44] | 0.627 | 0.660 | 0.590 | 0.442 | 0.433 | 0.555 |
| | *Baseline PPMd Compression-based Cosine* [19, 20] | 0.555 | 0.622 | 0.523 | 0.508 | 0.507 | 0.544 |
| 24 | Sun [45] | 0.525 | 0.622 | 0.506 | 0.499 | 0.498 | 0.531 |
| | *Baseline DetectLLM NPR (Mistral 7B)* [13] | 0.497 | 0.602 | 0.494 | 0.481 | 0.481 | 0.512 |
| 25 | Lei [46] | 0.598 | 0.604 | 0.604 | 0.318 | 0.378 | 0.504 |
| | *Baseline Fast-DetectGPT (Falcon 7B)* [15] | 0.480 | 0.626 | 0.474 | 0.457 | 0.458 | 0.500 |
| 26 | Liu [47] | 0.464 | 0.660 | 0.462 | 0.448 | 0.448 | 0.497 |
| 27 | *e-comm-tech*† | 0.463 | 0.651 | 0.467 | 0.445 | 0.446 | 0.497 |
| | *Baseline DetectGPT (Mistral 7B)* [14] | 0.472 | 0.552 | 0.476 | 0.468 | 0.465 | 0.488 |
| 28 | K. Huang [48] | 0.645 | 0.798 | 0.325 | 0.307 | 0.323 | 0.480 |
| | *Baseline DetectLLM NPR (Falcon 7B)* [13] | 0.445 | 0.575 | 0.449 | 0.432 | 0.433 | 0.468 |
| | *Baseline Authorship Unmasking* [21, 22] | 0.586 | 0.749 | 0.337 | 0.323 | 0.328 | 0.467 |
| 29 | Sheykhlan [49] | 0.627 | 0.789 | 0.304 | 0.282 | 0.296 | 0.460 |
| | *Baseline DetectLLM LRR (Falcon 7B)* [13] | 0.441 | 0.600 | 0.428 | 0.413 | 0.413 | 0.460 |
| 30 | G. Wu [50] | 0.493 | 0.586 | 0.409 | 0.366 | 0.382 | 0.450 |
| | *Baseline DetectGPT (Falcon 7B)* [14] | 0.409 | 0.526 | 0.425 | 0.413 | 0.412 | 0.439 |

* Scores estimated due to run failures on short texts. † No notebook submitted.

## 4.1. PAN Submission Ranking

We determined the final rank of each system by its macro-average mean *effectiveness* across all $n = 70$ dataset variants (including the five ELOQUENT submissions). Table 4 lists all systems sorted by their rank. Twelve of the systems beat the best baseline (Binoculars with Falcon 7B). Six more beat the second-best baseline (Binoculars with Mistral 7B). Figure 4 visualizes the mean score distribution of the systems as a boxplot. It can be seen that weaker systems not only have a lower mean score, but also a much higher variance over the different dataset variants. The bottom half of the systems even have worse-than-random scores on some variants, while performing quite well on others.

**Table 5**
PAN leader board considering only the unobfuscated *main* test collection. Systems are ranked by the mean of all evaluation measures across all other metrics discounted by half a standard deviation to correct for spread. The final ranking on all dataset variants is given in Table 4.

| | Team | ROC-AUC | Brier | C@1 | $F_1$ | $F_{0.5u}$ | Mean |
|---|---|---|---|---|---|---|---|
| 1 | Tavan [23] | **0.999** | **0.990** | **0.993** | **0.993** | **0.997** | **0.995** |
| 2 | Valdez-Valenzuela [36] | 0.985 | 0.985 | 0.985 | 0.985 | 0.983 | 0.985 |
| 3 | Zi. Lin [27] | 0.979 | 0.979 | 0.979 | 0.979 | 0.980 | 0.979 |
| 4 | J. Huang [24] | 0.980 | 0.980 | 0.980 | 0.979 | 0.977 | 0.979 |
| 5 | L. Guo [34] | 0.979 | 0.963 | 0.947 | 0.947 | 0.945 | 0.957 |
| 6 | W. Huang [39] | 0.955 | 0.955 | 0.955 | 0.954 | 0.954 | 0.955 |
| 7 | Miralles [29] | 0.972 | 0.929 | 0.955 | 0.954 | 0.954 | 0.953 |
| 8 | Abburi [28] | 0.979 | 0.945 | 0.943 | 0.940 | 0.943 | 0.950 |
| 9 | Lorenz [25] | 0.973 | 0.898 | 0.952 | 0.951 | 0.950 | 0.946 |
| | *Baseline Binoculars (Falcon 7B)* [12] | 0.943 | 0.928 | 0.926 | 0.920 | 0.922 | 0.928 |
| 10 | Gritsai [32] | 0.935 | 0.925 | 0.933 | 0.905 | 0.909 | 0.921 |
| 11 | M. Guo [26] | 0.915 | 0.911 | 0.920 | 0.919 | 0.916 | 0.916 |
| 12 | Yadagiri [30] | 0.961 | 0.871 | 0.916 | 0.893 | 0.899 | 0.908 |
| 13 | Ye [37] | 0.966 | 0.887 | 0.874 | 0.851 | 0.929 | 0.904 |
| 14 | Chen [38] | 0.885 | 0.886 | 0.886 | 0.898 | 0.856 | 0.882 |
| | *Baseline Binoculars (Mistral 7B)* [12] | 0.886 | 0.884 | 0.866 | 0.866 | 0.860 | 0.873 |
| 15 | B. Huang [35] | 0.866 | 0.878 | 0.861 | 0.856 | 0.856 | 0.863 |
| 16 | Z. Wu [42] | 0.907 | 0.854 | 0.809 | 0.807 | 0.813 | 0.838 |
| 17 | Lv [31] | 0.820 | 0.899 | 0.793 | 0.781 | 0.787 | 0.816 |
| | *Baseline Fast-DetectGPT (Mistral 7B)* [15] | 0.806 | 0.783 | 0.807 | 0.805 | 0.806 | 0.802 |
| 18 | Cao [33] | 0.798 | 0.798 | 0.798 | 0.796 | 0.797 | 0.797 |
| 19 | Qin [40] | 0.782 | 0.819 | 0.793 | 0.786 | 0.785 | 0.793 |
| 20 | Zhu [44] | 0.853 | 0.794 | 0.730 | 0.772 | 0.744 | 0.782 |
| 21 | Sheykhlan [49] | 0.807 | 0.858 | 0.674 | 0.641 | 0.656 | 0.727 |
| | *Baseline PPMd Compression-based Cosine* [19, 20] | 0.750 | 0.753 | 0.695 | 0.696 | 0.692 | 0.718 |
| 22 | K. Huang | 0.789 | 0.865 | 0.646 | 0.610 | 0.625 | 0.707 |
| 23 | Petropoulos [41] | 0.731 | 0.749 | 0.738 | 0.616 | 0.585 | 0.684 |
| 24 | Sun [45] | 0.646 | 0.744 | 0.649 | 0.646 | 0.647 | 0.667 |
| | *Baseline Authorship Unmasking* [21, 22] | 0.655 | 0.757 | 0.632 | 0.595 | 0.608 | 0.651 |
| | *Baseline DetectLLM-NPR (Mistral 7B)* [13] | 0.635 | 0.701 | 0.598 | 0.597 | 0.599 | 0.626 |
| | *Baseline Fast-DetectGPT (Falcon 7B)* [15] | 0.576 | 0.722 | 0.593 | 0.579 | 0.582 | 0.611 |
| 25 | Zh. Lin [43] | 0.624 | 0.624 | 0.624 | 0.548 | 0.628 | 0.610 |
| 26 | Liu [47] | 0.587 | 0.720 | 0.583 | 0.566 | 0.572 | 0.606 |
| | *Baseline DetectLLM-LRR (Mistral 7B)* [13] | 0.593 | 0.742 | 0.558 | 0.561 | 0.562 | 0.604 |
| 27 | Lei [46] | 0.631 | 0.630 | 0.630 | 0.508 | 0.612 | 0.602 |
| | Baseline Text Length | 0.601 | 0.600 | 0.600 | 0.604 | 0.603 | 0.602 |
| 28 | *gra*† | 0.500 | 0.750 | 0.488 | 0.656 | 0.544 | 0.587 |
| | *Baseline DetectLLM-NPR (Falcon 7B)* [13] | 0.584 | 0.680 | 0.559 | 0.551 | 0.550 | 0.585 |
| | *Baseline DetectLLM-LRR (Falcon 7B)* [13] | 0.558 | 0.704 | 0.539 | 0.533 | 0.536 | 0.575 |
| | *Baseline DetectGPT (Mistral 7B)* [14] | 0.550 | 0.671 | 0.532 | 0.533 | 0.532 | 0.564 |
| | *Baseline DetectGPT (Falcon 7B)* [14] | 0.493 | 0.663 | 0.489 | 0.487 | 0.487 | 0.525 |
| 29 | *e-comm-tech*† | 0.476 | 0.654 | 0.473 | 0.458 | 0.461 | 0.505 |
| 30 | G. Wu [50] | 0.487 | 0.574 | 0.427 | 0.384 | 0.402 | 0.456 |

† No notebook submitted.

In comparison, Table 5 shows the system rankings only on the more homogeneous and thus easier *main* test collection (without obfuscations and length restrictions, but with two German variants). In both rankings, Tavan and Najafi [23] ranks first, but the middle and lower ranks differ substantially, leading to an overall moderate to strong mean score rank correlation coefficient (Kendall's $\tau = .661$, $p \ll .001$). When measuring the main collection only, more systems outperform the lower baselines, but not the Binoculars baseline, which is actually beaten by fewer systems. Larger rank differences can also be seen among the top ranks. A notable example are Lorenz et al. [25], who rank ninth place on the main collection, but third place on all dataset variants. Conversely, Valdez-Valenzuela and Gómez-Adorno [36], who rank second on the main collection, rank only 14th on all dataset variants

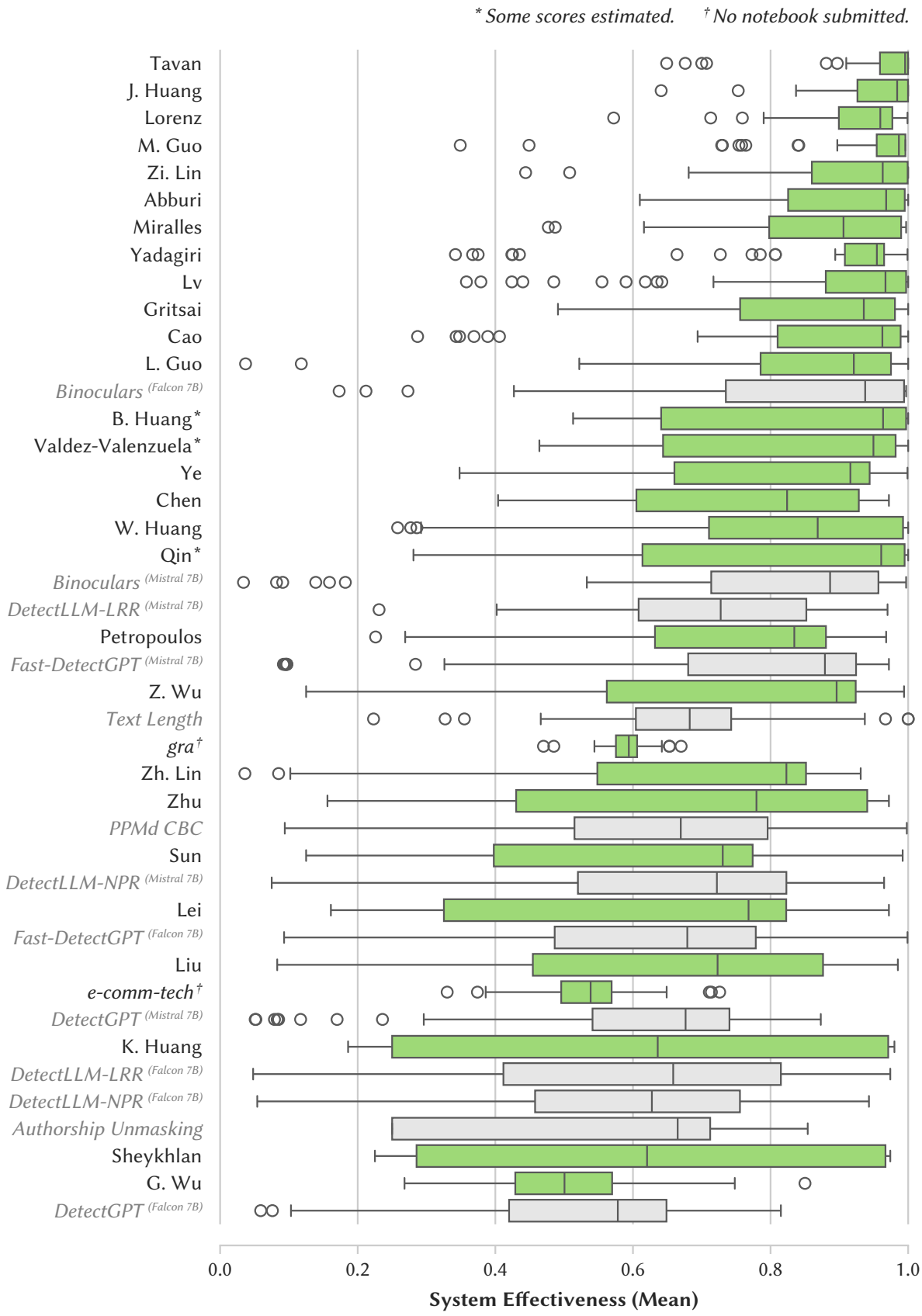


**Figure 4:** Mean score of all systems on each dataset variant in order of their rank (see Table 4). Baselines are marked in gray. Weaker systems tend to have not only a smal ler mean score, but also a much larger variance across different dataset variants.

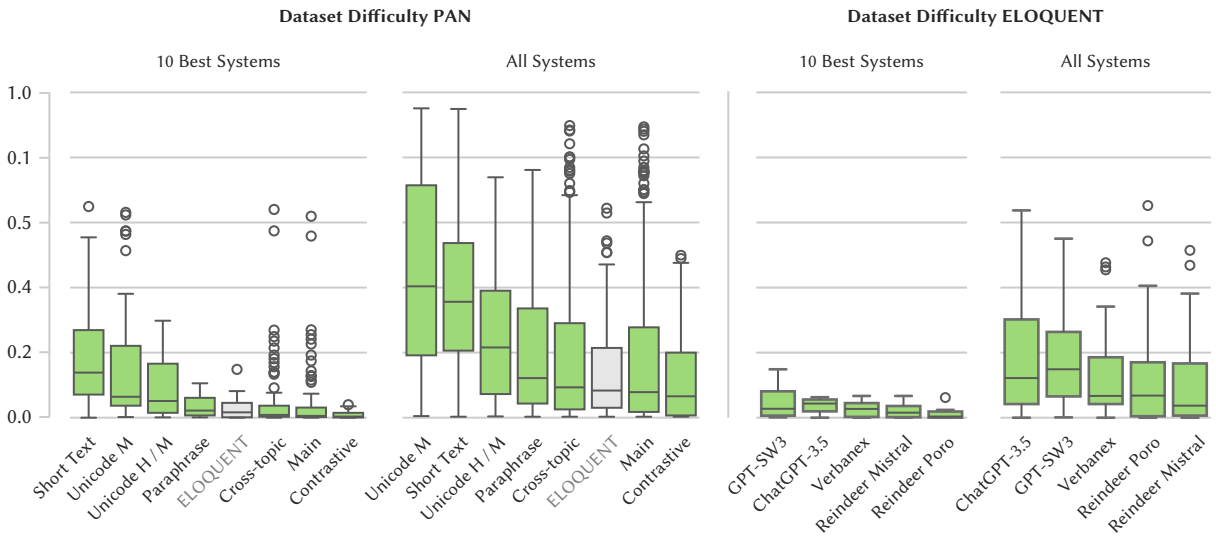


**Figure 5:** *Left:* Detection difficulty of all individual PAN test collections and the mean of all ELOQUENT submissions in comparison. *Right:* Detection difficulty of individual ELOQUENT submissions. The difficulty of a dataset is the inverse mean detection score of either the 10 best-performing systems or all systems. German texts are not shown separately, as they were part of the main collection. A more detailed comparison of individual dataset variants is given in Figure 6.

(although it should be noted that their short-text performance had to be estimated due to run failures). This shows that the top performance of many systems does not generalize well to unexpected or out-of-domain test cases. The top three systems [23, 24, 25], on the other hand, defy this trend and generalize quite well. Interestingly, all three use different approaches. Tavan and Najafi [23] built an ensemble with Binoculars and a fine-tuned Mistral LLM, Huang et al. [24] use a BERT classifier trained with PU loss [51] on sentence trigrams, and Lorenz et al. [25] employ a TF-IDF-based SVM classifier with surprisingly strong results.

## 4.2. PAN Dataset Variant Discussion

Figure 5 visualizes the difficulty of the PAN test collections. We determine the difficulty of a dataset by the inverse mean effectiveness score of the detection systems. Figure 6 shows in more detail the difficulty of all individual dataset variants that are part of the collections (in this case, based only on the scores of the ten best-performing systems).

Which of the other dataset variants was particularly difficult varied from system to system. Truncating texts to 35 words unsurprisingly yielded the most difficult dataset variant and all systems struggled with it. Three of systems even failed to run at all on these texts due to hard-coded assumptions in the software about the minimum length of a text. The effectiveness scores of these systems on the missing dataset variants was therefore estimated using the mean effectiveness score of all other systems. The three systems this affects [35, 40, 36] are marked with an asterisk in Table 4 and Figure 4.

Already part of the main test collection, but on its own very challenging for most systems was the set of German texts. Systems that relied on multilingual LLMs (such as Tavan and Najafi [23]) were less affected, but many other systems struggled. Interestingly, Lorenz et al. [25], achieved near-perfect scores on the German variant despite using a term-based SVM classifier with a fixed-length vocabulary trained on English texts. This suggests that the classifier fitted more on the typical characteristics of a human text rather than what makes a typical LLM text.

Of course, more difficult than short texts or German texts on their own were short German texts. Similarly, breaking the topic coherence (“cross-topic”) on its own was not very effective in making the dataset more difficult, but combining this condition with the German texts produced one of the most challenging variants (see Table Figure 6).

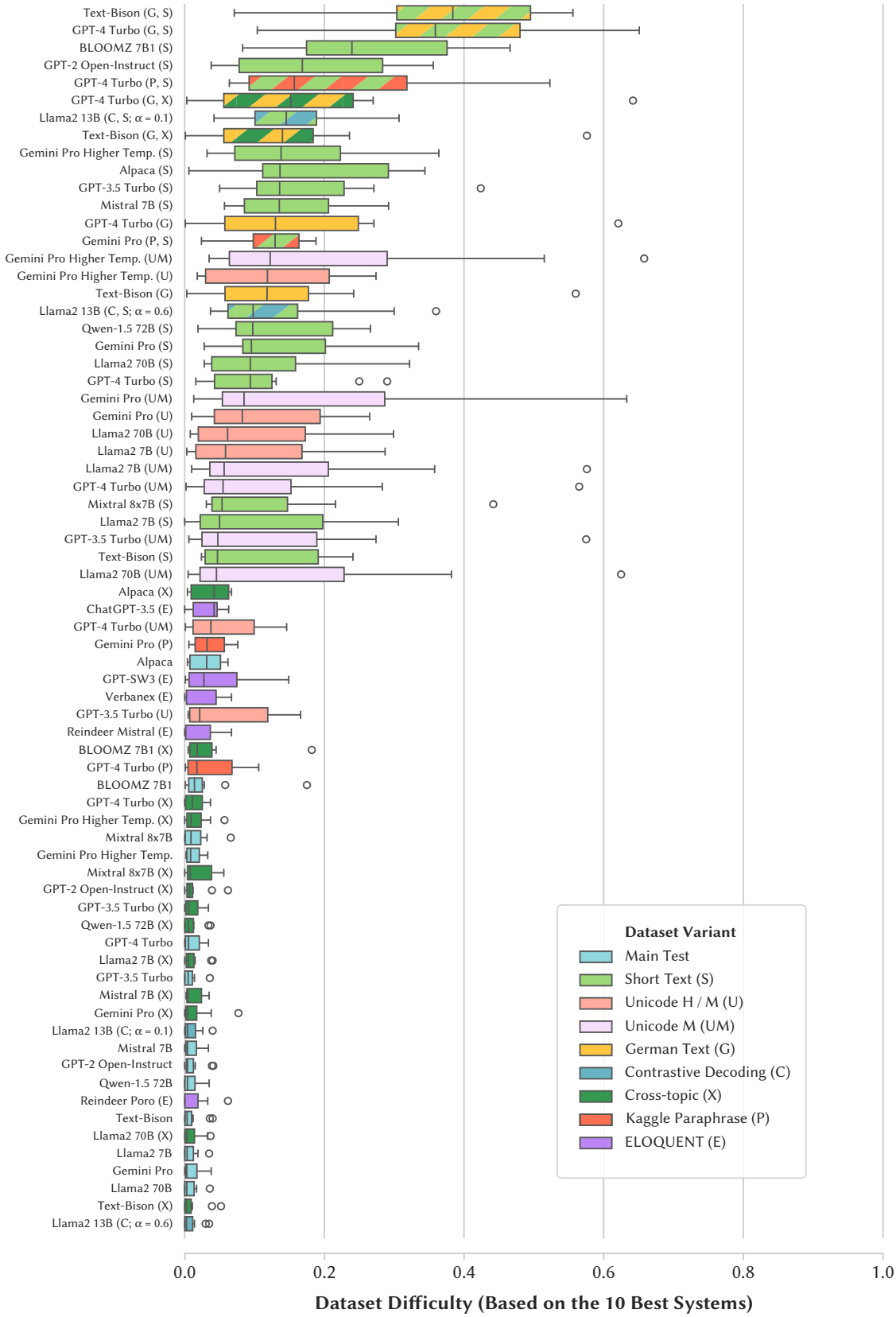


**Figure 6:** Detection difficulty of all individual LLMs and the dataset variants created from them according to the 10 best-performing systems. The unobfuscated main test set is shown in blue. Short texts are the most difficult followed by Unicode lookalike replacements applied to only the machine texts (M) or both human and machine texts (H M). The easiest-to-detect LLM is Llama2. GPT-4 and Gemini Pro are more difficult, though not by much.

**Table 6**
ELOQUENT leader board. The given scores are the mean inverse detection scores of the 10 best PAN systems.

| | Team | ROC-AUC$^{-1}$ | Brier$^{-1}$ | C@1$^{-1}$ | $F_1{}^{-1}$ | $F_{0.5u}{}^{-1}$ | Mean$^{-1}$ |
|---|---|---|---|---|---|---|---|
| 1 | GPT-SW3 | **0.023** | **0.064** | **0.051** | **0.056** | **0.051** | **0.049** |
| | *Baseline ChatGPT-3.5* | 0.021 | 0.056 | 0.032 | 0.042 | 0.034 | 0.037 |
| 2 | Verbanex | 0.012 | 0.048 | 0.024 | 0.031 | 0.025 | 0.028 |
| 3 | Reindeer Mistral | 0.009 | 0.046 | 0.020 | 0.024 | 0.017 | 0.023 |
| 4 | Reindeer Poro | 0.001 | 0.035 | 0.006 | 0.013 | 0.008 | 0.012 |

Unicode lookalike character replacements in the machine texts were another difficult obfuscation even for the best systems, whereas making the same replacements in the human texts as well equalized the effect to a certain degree. The resulting dataset variant in which both texts of a pair were obfuscated, was slightly less difficult than if only the machine text was obfuscated, yet still significantly more difficult than the unobfuscated original texts.

Contrastive decoding emerged as the easiest variant of all, even easier than the main dataset. However, this should be taken with a large grain of salt, since we used only Llama2 13B with two different settings for the hyper parameter $\alpha$ for creating these texts and Llama2 itself turned out to be the easiest LLM to detect (also see Figure 6).

### 4.3. ELOQUENT Submission Ranking

The ELOQUENT leader board is listed in Table 6. Of the four ELOQUENT submissions, only one submission (GPT-SW3) managed to beat the GPT-3.5 baseline in terms of dataset difficulty and all five (including the baseline) rank in the range of the PAN main test collection (see Figure 5 and Figure 6). None of the submissions proved more effective than any of the other PAN conditions, such as Unicode obfuscations or shortening the text length. This sobering result only goes to show how difficult to hide the fingerprints of current-generation LLMs still are without making drastic text modifications.

## 5. Conclusion

This year's shared task was designed as a builder-breaker task in collaboration with the ELOQUENT lab. PAN participants would *build* systems to detect LLM-generated texts and ELOQUENT participants would submit datasets trying to *break* the detectors.

In total, we received LLM detection systems from 30 different participants and evaluated them on 70 different dataset variants, five of which were contributed by ELOQUENT (including one baseline). Of the systems submitted to PAN, twelve beat the best of the provided baselines, demonstrating a robust performance across all dataset variants. The dataset variants were designed to test the generalization capabilities of the systems by exposing them to unseen and possibly unexpected conditions. We implemented this by replacing characters with Unicode lookalikes, shortening the texts to 35 words, generating German instead of English texts, and other similar means. Many of these variants proved quite challenging for the systems, though the best systems were able to handle most of them well.

ELOQUENT received four submissions, one of which beat the provided baseline. Unfortunately, none of the submissions proved effective enough to reduce the detection performance of the PAN systems.

We conclude that current LLMs are still easy to detect (some more than others) and their stylistic fingerprints are hard to hide. On the other hand, none of the detection systems managed to classify all test cases correctly, which means that despite all, there is a margin of error and we can expect this margin to increase with newer and better LLMs.

### Acknowledgments

The "Voight-Kampff" Generative AI Authorship Detection Task @ PAN 2024 has been funded as part of the OpenWebSearch project by the European Commission (OpenWebSearch.eu, GA 101070014).

## References


[1] A. M. Sarvazyan, J. Á. González, P. Rosso, M. Franco-Salvador, Supervised machine-generated text detectors: Family and scale matters, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction — 14th International Conference of the CLEF Association, Springer, 2023.

[2] J. King, P. Baffour, S. Crossley, R. Holbrook, M. Demkin, Llm – detect ai generated text, 2023. URL: https://kaggle.com/competitions/llm-detect-ai-generated-text.

[3] D. Molla, H. Zhan, X. He, Q. Xu, Overview of the 2023 ALTA shared task: Discriminate between human-written and machine-generated text, in: S. Muresan, V. Chen, K. Casey, V. David, D. Nina, I. Koji, E. Erik, U. Stefan (Eds.), Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association, Association for Computational Linguistics, 2023, pp. 148–152.

[4] A. 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, Procesamiento del Lenguaje Natural 71 (2023) 275–288. doi:10.48550/arXiv.2309.11285. arXiv:2309.11285.

[5] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, O. M. Afzal, T. Mahmoud, G. Puccetti, T. Arnold, SemEval-2024 Task 8: Multidomain, Multimodel and Multilingual Machine-Generated Text Detection, in: Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), 2024, pp. 2057–2079.

[6] E. Stamatatos, M. Potthast, F. M. R. Pardo, P. Rosso, B. Stein, Overview of the PAN/CLEF 2015 evaluation lab, in: J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. J. F. Jones, E. SanJuan, L. Cappellato, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings, volume 9283 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 518–538.

[7] J. Bevendorff, B. Ghanem, A. Giachanou, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. M. R. Pardo, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Wiegmann, E. Zangerle, Overview of PAN 2020: Authorship verification, celebrity profiling, profiling fake news spreaders on twitter, and style change detection, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, H. Joho, C. Lioma, C. Eickhoff, A. Névéol, L. Cappellato, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction - 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22-25, 2020, Proceedings, volume 12260 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 372–383.

[8] J. Bevendorff, B. Chulvi, G. L. D. la Peña Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle, Overview of PAN 2021: Authorship verification, profiling hate speech spreaders on twitter, and style change detection, in: K. S. Candan, B. Ionescu, L. Goeuriot, B. Larsen, H. Müller, A. Joly, M. Maistro, F. Piroi, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction - 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21-24, 2021, Proceedings, volume 12880 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 419–431.

[9] M. Kestemont, E. Manjavacas, I. Markov, J. Bevendorff, M. Wiegmann, E. Stamatatos, B. Stein, M. Potthast, Overview of the Cross-Domain Authorship Verification Task at PAN 2021, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), Working Notes Papers of the CLEF 2021 Evaluation Labs, volume 2936 of *CEUR Workshop Proceedings*, 2021. URL: https://ceur-ws.org/Vol-2936/paper-147.pdf.

[10] E. Stamatatos, M. Kestemont, K. Kredens, P. Pezik, A. Heini, J. Bevendorff, B. Stein, M. Potthast, Overview of the Authorship Verification Task at PAN 2022, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), CLEF 2022 Labs and Workshops, Notebook Papers, volume 3180 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3180/paper-184.pdf.

[11] E. Stamatatos, K. Kredens, P. Pezik, A. Heini, J. Bevendorff, B. Stein, M. Potthast, Overview of the Authorship Verification Task at PAN 2023, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos

(Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), volume 3497 of *CEUR Workshop Proceedings*, 2023, pp. 2476–2491. URL: https://ceur-ws.org/Vol-3497/paper-199.pdf.

[12] 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, arXiv [cs.CL] (2024). arXiv:2401.12070.

[13] J. Su, T. Y. Zhuo, D. Wang, P. Nakov, DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text, arXiv [cs.CL] (2023). arXiv:2306.05540.

[14] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, DetectGPT: Zero-shot machine-generated text detection using probability curvature, International Conference on Machine Learning 202 (2023) 24950–24962. doi:10.48550/arXiv.2301.11305. arXiv:2301.11305.

[15] G. Bao, Y. Zhao, Z. Teng, L. Yang, Y. Zhang, Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature, arXiv [cs.CL] (2023). arXiv:2310.05130.

[16] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, D. Hesslow, J. Launay, Q. Malartic, D. Mazzotta, B. Noune, B. Pannier, G. Penedo, The Falcon series of open language models, arXiv [cs.CL] (2023). arXiv:2311.16867.

[17] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7B, arXiv [cs.CL] (2023). arXiv:2310.06825.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv [cs.LG] (2019). arXiv:1910.10683.

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

[20] O. Halvani, C. Winter, L. Graner, On the usefulness of compression models for authorship verification, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, volume Part F1305, ACM, New York, NY, USA, 2017. doi:10.1145/3098954.3104050.

[21] M. Koppel, J. Schler, Authorship verification as a one-class classification problem, in: Twenty-first international conference on Machine learning - ICML '04, ACM Press, New York, New York, USA, 2004, pp. 489–495. doi:10.1145/1015330.1015448.

[22] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Generalizing unmasking for short texts, in: Proceedings of the 2019 Conference of the North, Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 654–659. doi:10.18653/v1/n19-1068.

[23] E. Tavan, M. Najafi, Marsan at PAN: BinocularLLM and Fusing Binoculars' Insight with the Proficiency of Large Language Models for Cutting-Edge Machine-Generated Text Detection, 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-WS.org, 2024.

[24] J. Huang, Y. Chen, M. Luo, Y. Li, Generative AI Authorship Verification Of Tri-Sentence Analysis Base On The Bert Model, 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-WS.org, 2024.

[25] L. Lorenz, F. Z. Aygüler, F. Schlatt, N. Mirzakhmedova, BaselineAvengers at PAN 2024: Often-Forgotten Baselines for LLM-Generated Text Detection, 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-WS.org, 2024.

[26] M. Guo, Z. Han, H. Chen, J. Peng, A Machine-Generated Text Detection Model Based on Text Multi-Feature Fusion, 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-WS.org, 2024.

[27] Z. Lin, Z. Han, L. Kong, M. Chen, S. Zhang, J. Peng, K. Sun, A Verifying Generative Text Authorship Model With Regularized Dropout, 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-WS.org,

2024.
[28] H. Abburi, N. Pudota, B. Veeramani, E. Bowen, S. Bhattacharya, Team Deloitte at PAN: Generative AI Text Detection, 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-WS.org, 2024.
[29] P. Miralles, A. Martín, D. Camacho, Ensembling Normalized Log Probabilities, 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-WS.org, 2024.
[30] A. Yadagiri, D. Kalita, A. Ranjan, A. Bostan, P. Toppo, P. Pakray, Team cnlp-nits-pp at PAN: Leveraging BERT for Accurate Authorship Verification: A Novel Approach to Textual Attribution, 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-WS.org, 2024.
[31] J. Lv, Y. Han, L. Kong, Meta-Contrastive Learning for Generative AI Authorship Verification, 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-WS.org, 2024.
[32] G. Gritsai, G. Boyeva, A. Grabovoy, Team ap-team at PAN: LLM Adapters for Various Datasets, 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-WS.org, 2024.
[33] H. Cao, Z. Han, J. Ye, B. Liu, Y. Han, Enhancing Human-Machine Authorship Discrimination in Generative AI Verification Task with BERT and Augmented Data, 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-WS.org, 2024.
[34] L. Guo, W. Yang, L. Ma, J. Ruan, BLGAV: Generative AI Author Verification Model Based on BERT and BiLSTM, 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-WS.org, 2024.
[35] B. Huang, C. Zhong, K. Yan, Y. Han, Author authentication of generative AI based on BERT by regularization method, 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-WS.org, 2024.
[36] A. Valdez-Valenzuela, H. Gómez-Adorno, Team iimasnlp at PAN: Leveraging Graph Neural Networks and Large Language Models for Generative AI Authorship Verification, 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-WS.org, 2024.
[37] Z. Ye, Y. Zhong, Z. Huang, L. Kong, Token Prediction as Implicit Classification for Generative AI Authorship Verification, 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-WS.org, 2024.
[38] J. Chen, L. Kong, Integrating Dual BERT Models and Causal Language Models for Enhanced Detection of Machine-Generated Texts, 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-WS.org, 2024.
[39] W. Huang, J. Grieve, Authorial Language Models For AI Authorship Verification, 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-WS.org, 2024.
[40] R. Qin, H. Qi, Y. Yi, A model fusion approach for generative AI authorship verification, 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-WS.org, 2024.
[41] P. Petropoulos, V. Petropoulos, RoBERTa and Bi-LSTM for Human vs AI generated Text Detection, 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-WS.org, 2024.
[42] Z. Wu, W. Yang, L. Ma, Z. Zhao, BertT: A Hybrid Neural Network Model for Generative AI Authorship Verification, 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-WS.org, 2024.
[43] Z. Lin, Y. Li, J. Huang, Voight-Kampff Generative AI Authorship Verification Based on T5, 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-WS.org, 2024.
[44] Y. Zhu, L. Kong, AI Authorship Verification Based On Deberta Model, 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-WS.org, 2024.
[45] G. Sun, W. Yang, L. Ma, BCAV: A Generative AI Author Verification Model Based on the Integration of Bert and CNN, 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-WS.org, 2024.
[46] H. Lei, X. Liu, G. Niu, Y. Zhou, Y. Zhou, Generative AI Authorship Verification based on ChatGLM, 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-WS.org, 2024.
[47] X. Liu, L. Kong, AI Text Detection Method Based on Perplexity Features with Strided Sliding Window, 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-WS.org, 2024.
[48] K. Huang, H. Qi, K. Yan, Voight-Kampff Generative AI Authorship Verification based on Contrastive Learning and Domain Adaptation, 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-WS.org, 2024.
[49] M. Sheykhlan, S. Abdoljabbar, M. Mahmoudabad, Team karami-kheiri at PAN: Enhancing Machine-Generated Text Detection with Ensemble Learning Based on Transformer Models, 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-WS.org, 2024.
[50] G. Wu, Q. Guan, Team lm-detector at PAN: Can NLI be an Appropriate Approach to Machine-Generated Text Detection, 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-WS.org, 2024.
[51] Y. Tian, H. Chen, X. Wang, Z. Bai, Q. Zhang, R. Li, C. Xu, Y. Wang, Multiscale positive-unlabeled detection of ai-generated texts, CoRR abs/2305.18149 (2023). URL: https://doi.org/10.48550/arXiv.2305.18149. doi:10.48550/ARXIV.2305.18149.
[52] X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T. Liu, R-drop: Regularized dropout for neural networks, in: M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, 2021, pp. 10890–10905.
[53] A. Nichol, J. Achiam, J. Schulman, On first-order meta-learning algorithms, arXiv [cs.LG] (2018). arXiv:1803.02999.
[54] S. Thite, Llm - detect ai generated text dataset, 2023. URL: https://www.kaggle.com/datasets/sunilthite/llm-detect-ai-generated-text-dataset.
[55] D. Kłeczek, Daigt v2 train dataset, 2023. URL: https://www.kaggle.com/datasets/thedrcat/daigt-v2-train-dataset.
[56] R. Luukkonen, J. Burdge, E. Zosa, A. Talman, V. Komulainen, V. Hatanpää, P. Sarlin, S. Pyysalo, Poro 34b and the blessing of multilinguality, arXiv preprint: 2404.01856 (2024).
[57] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7b, arXiv preprint arXiv:2310.06825 (2023).
[58] V. Neralla, S. Bijl de Vroe, Evaluating Poro-34B-Chat and Mistral-7B-Instruct-v0.1: LLM System Description for ELOQUENT at CLEF 2024, in: G. Faggioli, N. Ferro, M. Vlachos, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2024.
[59] A. Ekgren, A. Cuba Gyllensten, F. Stollenwerk, J. Öhman, T. Isbister, E. Gogoulou, F. Carlsson, J. Casademont, M. Sahlgren, GPT-SW3: An autoregressive language model for the Scandinavian languages, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italia, 2024.
[60] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh,

D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, arXiv [cs.CL] (2020). arXiv:2005.14165.

[61] OpenAI, GPT-4 Technical Report, arXiv [cs.CL] (2023). arXiv:2303.08774.

[62] Gemini Team, Gemini: A family of highly capable multimodal models, arXiv [cs.CL] (2023). arXiv:2312.11805.

[63] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, E. Chu, J. H. Clark, L. E. Shafey, Y. Huang, K. Meier-Hellstern, G. Mishra, E. Moreira, M. Omernick, K. Robinson, S. Ruder, Y. Tay, K. Xiao, Y. Xu, Y. Zhang, G. H. Abrego, J. Ahn, J. Austin, P. Barham, J. Botha, J. Bradbury, S. Brahma, K. Brooks, M. Catasta, Y. Cheng, C. Cherry, C. A. Choquette-Choo, A. Chowdhery, C. Crepy, S. Dave, M. Dehghani, S. Dev, J. Devlin, M. Díaz, N. Du, E. Dyer, V. Feinberg, F. Feng, V. Fienber, M. Freitag, X. Garcia, S. Gehrmann, L. Gonzalez, G. Gur-Ari, S. Hand, H. Hashemi, L. Hou, J. Howland, A. Hu, J. Hui, J. Hurwitz, M. Isard, A. Ittycheriah, M. Jagielski, W. Jia, K. Kenealy, M. Krikun, S. Kudugunta, C. Lan, K. Lee, B. Lee, E. Li, M. Li, W. Li, Y. Li, J. Li, H. Lim, H. Lin, Z. Liu, F. Liu, M. Maggioni, A. Mahendru, J. Maynez, V. Misra, M. Moussalem, Z. Nado, J. Nham, E. Ni, A. Nystrom, A. Parrish, M. Pellat, M. Polacek, A. Polozov, R. Pope, S. Qiao, E. Reif, B. Richter, P. Riley, A. C. Ros, A. Roy, B. Saeta, R. Samuel, R. Shelby, A. Slone, D. Smilkov, D. R. So, D. Sohn, S. Tokumine, D. Valter, V. Vasudevan, K. Vodrahalli, X. Wang, P. Wang, Z. Wang, T. Wang, J. Wieting, Y. Wu, K. Xu, Y. Xu, L. Xue, P. Yin, J. Yu, Q. Zhang, S. Zheng, C. Zheng, W. Zhou, D. Zhou, S. Petrov, Y. Wu, PaLM 2 Technical Report, arXiv [cs.CL] (2023). arXiv:2305.10403.

[64] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, LLaMA: Open and efficient foundation language models, arXiv [cs.CL] (2023). arXiv:2302.13971.

[65] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mixtral of Experts, arXiv [cs.LG] (2024). arXiv:2401.04088.

[66] N. Muennighoff, T. Wang, L. Sutawika, A. Roberts, S. Biderman, T. L. Scao, M. S. Bari, S. Shen, Z.-X. Yong, H. Schoelkopf, X. Tang, D. Radev, A. F. Aji, K. Almubarak, S. Albanie, Z. Alyafeai, A. Webson, E. Raff, C. Raffel, Crosslingual Generalization through Multitask Finetuning, arXiv [cs.CL] (2022). arXiv:2211.01786.

[67] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, T. Zhu, Qwen Technical Report, arXiv [cs.CL] (2023). arXiv:2309.16609.

[68] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, HuggingFace's transformers: State-of-the-art natural language processing, arXiv [cs.CL] (2019). arXiv:1910.03771.

[69] X. L. Li, A. Holtzman, D. Fried, P. Liang, J. Eisner, T. Hashimoto, L. Zettlemoyer, M. Lewis, Contrastive decoding: Open-ended text generation as optimization, arXiv [cs.CL] (2022). arXiv:2210.15097.

[70] Y. Wang, H. Ivison, P. Dasigi, J. Hessel, T. Khot, K. R. Chandu, D. Wadden, K. MacMillan, N. A. Smith, I. Beltagy, H. Hajishirzi, How far can camels go? Exploring the state of instruction tuning on open resources, Neural Information Processing Systems abs/2306.04751 (2023) 74764–74786. doi:10.48550/arXiv.2306.04751. arXiv:2306.04751.

[71] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford alpaca: An instruction-following llama model, https://github.com/tatsu-lab/stanford_alpaca, 2023.

[72] Y. Su, T. Lan, Y. Wang, D. Yogatama, L. Kong, N. Collier, A contrastive framework for neural text generation, arXiv [cs.CL] (2022). arXiv:2202.06417.