# Synthetic Lyrics Detection Across Languages and Genres

**Yanis Labrak** [1,2] **Markus Frohmann** [1,3] **Gabriel Meseguer-Brocal** [1] **Elena V. Epure** [1]
Deezer Research, Paris, France [1]
LIA - Avignon University, Avignon, France [2] Johannes Kepler University Linz, Austria [3]
research@deezer.com

## Abstract

In recent years, the use of large language models (LLMs) to generate music content, particularly lyrics, has gained in popularity. These advances provide valuable tools for artists and enhance their creative processes, but they also raise concerns about copyright violations, consumer satisfaction, and content spamming. Previous research has explored content detection in various domains. However, no work has focused on the text modality, lyrics, in music. To address this gap, we curated a diverse dataset of real and synthetic lyrics from multiple languages, music genres, and artists. The generation pipeline was validated using both humans and automated methods. We performed a thorough evaluation of existing synthetic text detection approaches on lyrics, a previously unexplored data type. We also investigated methods to adapt the best-performing features to lyrics through unsupervised domain adaptation. Following both music and industrial constraints, we examined how well these approaches generalize across languages, scale with data availability, handle multilingual language content, and perform on novel genres in few-shot settings. Our findings show promising results that could inform policy decisions around AI-generated music and enhance transparency for users.

## 1 Introduction

Recent advancements in user-friendly tools, such as Suno AI[1], have significantly impacted the music field by introducing prompt-based interfaces that simplify music generation. In parallel, multiple research works have been exploring audio generation (Agostinelli et al., 2023; Dhariwal et al., 2020; Wu et al., 2024) or lyrics generation (Qian et al., 2023; Nikolov et al., 2020; Tian et al., 2023) with impressive results. LLMs such as GPT-4 (OpenAI et al., 2024b), Mistral 7B (Jiang et al., 2023), Gemma (Mesnard et al., 2024), or PaLM (Chowdhery et al., 2022) have demonstrated the ability to generate human-like text without adaptation, being able to assist artists in tasks such as poem writing (Popescu-Belis et al., 2023) and song lyrics creation (Qian et al., 2023).

Nevertheless, the widespread use of LLMs for generating artistic content has raised concerns regarding authorship infringement (Novelli et al., 2024; Goetze, 2024), consumer satisfaction (ChrissyGee et al., 2024), and content spamming. These concerns outline the need to effectively detect synthetic content to regulate its distribution and prevent misuse. Although many methods for synthetic text detection have been proposed and explored (Abburi et al., 2023; Chen et al., 2023; Wu et al., 2023; Pu et al., 2023; Wang et al., 2024; Dugan et al., 2024; Li et al., 2024), their effectiveness in detecting AI-generated lyrics as a form of creative content remains unclear. Lyrics differ significantly from other text types due to their unique semantics, rhythmic structures, and socio-cultural references (Spanu, 2019). Also, existing detection benchmarks predominantly focus on English, limiting their applicability across languages, and the synthetic text used in these evaluations is often not rigorously validated. To overcome these limitations, we propose the following contributions:

- We carefully design a generation and post-processing pipeline to produce realistic lyrics, which we then validate through a human study and with automatic methods.
- We create and release a dataset of synthetic lyrics by using multiple generative models, featuring a wide range of lyrics for 9 languages and 18 unique music genres inspired by 1,771 artists from various countries.
- We conduct extensive experiments to benchmark existing text detection approaches on this new type of synthetic text (creative and multilingual) with minimal adaptation. Our focus includes a variety of features: metrics derived from per-token probabilities in lyrics and stylistic and sentence embeddings. Then,

[1] suno.com

we assess LLM2Vec (BehnamGhader et al., 2024) for the first time in the context of text detection, both with and without lyrics-specific adaptation, showing that it outperforms all other features on this data type.

- In contrast to previous works, we evaluate detectors not only for generalization to unseen generators and content (e.g., new artist style, new music genres) but also for their robustness and performance with unseen languages and varying levels of data availability in order to simulate a more realistic detection scenario.

Data, pre-processing scripts, code, and models will be publicly accessible on GitHub [2] under the Apache 2.0 license and in compliance with the content copyrights.

## 2 Related Work

The detection of machine-generated content has emerged as a well-established research domain (Lavergne et al., 2008; Badaskar et al., 2008; Yang et al., 2023; Rana et al., 2022; Ahmed et al., 2022; Zhou and Lim, 2021; Guarnera et al., 2024; Bammey, 2024). Traditionally, efforts have focused on identifying generated text in areas like news (Bhat and Parthasarathy, 2020; Schuster et al., 2020), scientific writing (Chen et al., 2021), or voice spoofing in audio (Wu et al., 2017a; Zhang et al., 2021). However, recent advances in generative models in terms of quality and creativity have underscored the need for detectors capable of identifying more complex forms of machine-generated text, such as creative content. In music, multiple modalities are vulnerable to AI-generated content, but current efforts have mainly targeted audio detection (Zang et al., 2024; Wu et al., 2017b; Afchar et al., 2024).

Detection of machine-generated text is typically framed as a binary classification task distinguishing between human-written and synthetic content (Liu et al., 2023; Huang et al., 2024). One way of solving it relies on supervised learning, where classifiers are trained based on textual encoders like RoBERTa or Longformer (Abdelnabi and Fritz, 2021; Chakraborty et al., 2023; Kirchenbauer et al., 2023; Liu et al., 2023; Wang et al., 2024; Li et al., 2024) or LLMs (Macko et al., 2023; Antoun et al., 2024; Chen et al., 2023; Kumarage et al., 2023). This approach requires a sufficiently large training corpus, which is not always available, and may encounter overfitting issues on unseen data, including new authorial styles or generative models (Uchendu et al., 2020; Bakhtin et al., 2019).

[2] https://github.com/deezer/synthetic_lyrics_detection

Another line of research has focused on distinguishing between machine-generated and human-written texts using various metrics derived from output probabilities of generative models or stylistic features (Mitchell et al., 2023; Su et al., 2023; Zhu et al., 2023; Sadasivan et al., 2024; Soto et al., 2024). These methods have been proven effective, while sometimes shown to yield lower performance than the supervised ones depending on the generative model and data (Wang et al., 2024; Li et al., 2024). Parallel research has explored watermark-based detection methods (Abdelnabi and Fritz, 2021; Chakraborty et al., 2023; Kirchenbauer et al., 2023), but these approaches are limited by the requirement to access model logits, which is not feasible for models accessible only via APIs, such as GPT-4 (OpenAI et al., 2024a).

As discussed above, previous research has explored content detection across various domains, yet no work has exclusively focused on the text modality, lyrics, in music. Moreover, prior benchmarks have primarily targeted English text and often lacked a rigorous validation of the synthetic text used in experiments, raising concerns about the findings' reliability and generalization. These gaps highlight the need for a validated pipeline to generate and refine lyrics, the release of synthetic data that is realistic, musically diverse, and multilingual, and more targeted generalization experiments that explore various factors, including generative models, languages, and writing styles.

## 3 Data Creation and Validation

As no prior public studies have addressed the detection of machine-generated lyrics, there is a lack of data reflecting the inherent diversity of song lyrics. To address this gap, we introduce and document the creation of the first lyrics dataset specifically designed for synthetic lyrics detection. This data encompasses a wide variety of artistic styles, music genres, and languages. For generation, we chose to focus on textual input only, excluding lyrics generators that use multiple modalities, such as melody or audio (Qian et al., 2023; Tian et al., 2023). Likewise, we align with the most widely used tools among content creators, such as Suno and ChatGPT, which produce lyrics based entirely on text.

### 3.1 Human-Written Lyrics Dataset

Given the large diversity of the music catalog with lyrics from millions of artists across very different genres, styles, and languages, with new tracks being added almost every second (Ingham, 2021), creating a comprehensive dataset that covers these

dimensions is necessary but challenging.

For this work, we curated a multilingual dataset of 3,704 human-written lyrics targeting nine languages: English (EN), German (DE), Turkish (TR), French (FR), Portuguese (PT), Spanish (ES), Italian (IT), Arabic (AR), and Japanese (JA). The inclusion criterion was based on popularity, specifically from tracks listed in the most popular editorial playlists on an international music streaming platform[3] as of June 2024. Also, we ensured that each track was released within the past year and a half to minimize the possibility that the models used in the detectors had prior exposure to this content. We evenly selected lyrics only from top-trending music genres per language, as determined by daily streaming statistics at extraction time. Appendix A shows the data distribution, and Appendix B the list of popular genres per language.

To allow a quality assessment of the generated lyrics by English-speaking humans from our organization, we decided to evenly and randomly pick a sub-sample from this dataset focused on the five most popular artists from the 2023 Billboard "Top Artists"[4], namely: Drake, Ed Sheeran, Post Malone, Taylor Swift, and The Weeknd. Though limited in scope, this dataset is a test bed of 625 human-written lyrics (for the distribution, see Appendix A) well-suited for assessing artistic style cloning capabilities of our LLM generation pipeline. We also use this controlled subset to identify the best detection features before running extensive experiments on robustness, scalability, and generalization.

### 3.2 Synthetic Lyrics Dataset

High-quality generated text increases the difficulty of the task, providing a better evaluation and insights into a system's ability to generalize to unseen data. To produce human-like lyrics, we designed a four-step process that was refined through multiple iterations, with each step's output being empirically evaluated for potential issues or generation artifacts and improvements made accordingly. The entire pipeline is validated through a human study (Section 3.3) and an automatic evaluation focused on the regurgitation of the models (Section 3.4).

**Step 1 - Generation.** We opted for a constrained generation with a carefully designed prompt that was short and general, including some basic formatting instructions and three lyrics examples. The few-shot examples changed at each generation to diversify the output (Lu et al., 2022) but were conditioned on the same artist for the Billboard top artists data or the same language/genre pair for the multilingual data. To ensure the generated lyrics closely resembled real ones, the model was instructed to follow the same formatting guidelines as the real lyrics[5]. Appendix C shows the prompt template and Appendix D the hyperparameters used.

We selected four LLMs to generate varied content, ensuring their release preceded the period of the human-written lyrics. LLaMa 2 13B (Touvron et al., 2023) and Mistral 7B (Jiang et al., 2023) were chosen as the foundation models. In particular, lyrics generated with LLaMa 2 13B were used only as training data for the Billboard top artist subset to validate generalization capabilities to new models. TinyLLaMa 1.1B (Zhang et al., 2024) was used as a smaller, more compact model with similar performance to its corresponding foundation model. Lastly, we included WizardLM2 7B (Xu et al., 2024), an instruction-tuned model derived from Mistral 7B and fine-tuned on a large dataset using DPO (Rafailov et al., 2023).

**Step 2 - Normalization.** We normalized generated lyrics using regular expressions developed iteratively with each model's inclusion to remove artifacts not found in real lyrics, such as punctuation at the end of verses, quotations, references to the generation process (e.g., "here's an example of a song"), and indications of offensive content.

**Step 3 - Initial Filtering.** We sampled normalized generated lyrics to match the typical style of artists or language/genre pairs using statistical metrics from real lyrics, such as sentence length, number of verses, verse size, and word count. Only lyrics that fell within the interquartile range of these metrics, represented by box plots created from the human-written lyrics per artist, were retained.

**Step 4 - Semantic Similarity Filtering.** We performed a semantic similarity comparison between generated and human-written lyrics, retaining up to 150 synthetic lyrics that were most similar for each generative model and artist or language-genre pair. For this, we used the Sentence Transformers's (Reimers and Gurevych, 2019) model `all-MiniLM-L6-v2` from Wang et al. (2021).

### 3.3 Human Evaluation

The human evaluation aimed to assess how realistic the lyrics produced by our generation and post-processing pipeline were, providing insights into their validity. We recruited four English-speaking subjects from our organization to determine whether 70 English lyrics from the Billboard

[3] deezer.com

[4] billboard.com/charts/year-end/top-artists

[5] docs.lyricfind.com

top artists data were 'human-written' or 'machine-generated', based on text only. The samples were evenly split between the two classes and uniformly distributed across various artists and generative models, while subjects were unaware of this distribution to prevent bias. Subjects also rated their confidence in each annotation on a scale from 1 to 4 (details in Appendix E). Post-annotation, an unstructured interview was conducted to gather insights into the decision-making process (e.g., cues used in judgments), familiarity with the lyrics, and perceived difficulty (transcribed in Appendix G).

Table 1 shows that the differences among subjects are substantial, with a gap of 36.9 points between the highest (ID 4) and lowest (ID 2) scores. The recall for the synthetic lyrics is close to or even worse than a random baseline for all the subjects except the fourth. The detection of human-written lyrics appears better, but this might be related to a tendency to overuse this label in annotation.

| *Subject ID* | *Synthetic* | *Human-written* | *Overall* |
|---|---|---|---|
| *1* | 54.3 | 97.1 | 75.7 |
| *2* | 40.0 | 43.4 | 41.7 |
| *3* | 57.1 | 78.5 | 67.8 |
| *4* | 74.3 | 82.9 | 78.6 |

Table 1: Human subjects' recall on a sample of 70 lyrics taken from the Billboard top artists data.

In Appendix F, we show that subjects tended to assign slightly lower confidence scores to their incorrect annotations, likely because they anticipated their mistakes to some extent. Based on subjects' feedback detailed in Appendix G, only one popular song by Taylor Swift was recognized. We provide a supplementary analysis of pair inter-rater agreements in Appendix F. Overall, the results highlight the task's difficulty and that the generated lyrics resemble real ones, thus validating our pipeline.

### 3.4 Measuring Few-Shot Regurgitation

To ensure that the generative models used for creating our dataset do not merely reproduce the provided few-shot examples, we conducted an additional evaluation of the generated lyrics apart from the human one. We indexed all the human-written lyrics used to condition the models in generation with the BM25 representation (Trotman et al., 2014). Then, we queried this corpus by using synthetic lyrics and checked if the few-shot examples provided as seeds in the corresponding prompt during generation scored high in this retrieval task. Table 2 shows that hit rates are relatively low for each rank range, indicating a low likelihood of the generated lyrics being based on the set of lyrics provided as input to condition their generation.

| Rank | % Hit rate | Cumulated % Hit rate |
|---|---|---|
| 1 | 2.28 | 2.28 |
| 2 | 1.05 | 3.34 |
| 3 | 0.83 | 4.17 |
| 3 to 5 | 1.37 | 5.55 |
| 5 to 10 | 2.57 | 8.12 |
| 10 to 20 | 3.94 | 12.06 |
| 20 to 50 | 7.79 | 19.86 |

Table 2: Hit rate (%) by rank range when retrieving the human lyrics used as 3-shot examples during generation with the corresponding synthetic lyrics.

## 4 Lyrics Detection Experiments

We approached the detection task as a few-shot prediction using a k-nearest neighbors (k-NN) algorithm on a pre-computed lyrics features space. This method, which works with a limited set of lyrics, supports continuous updates as new synthetic content, including human-flagged material, becomes available. The vector space is constructed using both human-written and machine-generated lyrics, corresponding to our binary classification setup, incorporating multiple features commonly used in text detection (as detailed in Section 4.2). During evaluation, we applied a distance-based metric (Minkowski) to find the $k$ closest points to the input and assign the most frequent label (with $k = 3$ in our experiments). This approach also allowed for better control and explainability by understanding the influence of individual features[6].

### 4.1 Data Split and Evaluation Scenarios

**Billboard Top Artists Detection.** We extended the 625 human-written lyrics of the Billboard top artists data with 4,572 synthetic lyrics inspired by the same artists. To evaluate cross-artist and cross-model generalization, we reserved the lyrics from two out of five artists (The Weeknd and Taylor Swift) exclusively to assess the detector's ability to generalize to unseen authorial styles. The lyrics from the other artists were used for both training and evaluation splits. For training, we sampled 300 lyrics, evenly split between human-written and machine-generated (50 lyrics from each artist).

*Cross-Artist and Cross-Model Generalization.* We aimed to first assess the generalization capabilities to unseen generative models (Mistral 7B, TinyLLaMa, and WizardLM2) and new artists (Taylor Swift and The Weeknd, as previously detailed).

[6]While k-NN is susceptible to feature scaling, this does not pose a problem since we have full control over the features.

| | *Lyrics Generators* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Mistral 7B | | TinyLLaMa | | WizardLM2 | | Human-written | | |
| | *S* | *U* | *S* | *U* | *S* | *U* | *S* | *U* | Avg. |
| *Random* | 51.3 | 49.0 | 50.2 | 48.7 | 46.9 | 53.3 | 48.0 | 41.3 | 47.3 |
| *Metrics based on LLaMa 2 7B Per-Tokens Probabilities* | | | | | | | | | |
| *Perplexity* | 79.0 | 84.0 | 58.0 | 45.3 | 71.9 | 72.7 | 57.2 | 53.6 | 61.9 |
| *Max.Neg Log.Lkl.* | 75.8 | 74.3 | **77.6** | **72.3** | 63.2 | 55.7 | 83.4 | 89.4 | 78.1 |
| *Shannon Entropy* | | | | | | | | | |
| *Max* | 88.2 | 94.0 | 50.6 | 58.9 | 71.6 | 73.0 | 77.4 | 71.2 | 73.5 |
| *Max+Min* | 88.4 | 88.7 | 64.6 | 60.2 | 68.6 | 65.3 | 80.6 | 82.8 | 77.2 |
| *Min-K%Prob (K=10)* | 92.4 | 93.7 | 70.5 | 51.0 | **93.2** | **96.7** | 70.7 | 88.6 | 81.3 |
| *Semantic and Syntactic Embeddings* | | | | | | | | | |
| *SBERT* | | | | | | | | | |
| *MiniLMv2* | 86.9 | 94.3 | 54.7 | 55.2 | 87.9 | 91.7 | 74.8 | 73.5 | 76.3 |
| *MPNet* | 86.4 | 95.7 | 52.0 | 51.2 | 88.5 | 92.7 | 82.3 | 79.7 | 79.4 |
| *LLM2vec* | | | | | | | | | |
| *LLaMa3 8B* | **95.1** | **96.7** | 70.0 | 59.4 | 78.3 | 80.0 | 94.7 | 95.6 | **87.5** |
| *LLaMa2 7B* | 77.8 | 88.0 | 57.5 | 45.3 | 45.1 | 48.3 | **97.6** | 90.8 | 77.3 |
| *Stylistic Embeddings* | | | | | | | | | |
| *UAR* | | | | | | | | | |
| *CRUD* | 74.7 | 81.0 | 32.8 | 32.9 | 44.8 | 44.7 | 90.6 | 89.1 | 70.8 |
| *MUD* | 84.2 | 88.0 | 32.7 | 37.4 | 53.2 | 59.0 | 95.4 | **95.7** | 77.3 |

Table 3: Recall scores on the Billboard top artists dataset based on various features. $S$ refers to artists seen in the vector space, and $U$ to the unseen ones. Avg. is the overall micro recall between human-written and synthetic classes. For each feature category, the best-performing one is in **bold**, and the second-best is underlined.

**Multilingual Lyrics Detection.** The dataset consists of 7,262 lyrics, with 3,558 being synthetic and 3,704 human-written, distributed across 1,771 unique artist styles. For training, we randomly sampled up to 5 lyrics for each class (human-written and synthetic) and each language/genre pair. The remaining lyrics were reserved for evaluation. The distribution across splits is shown in Appendix A. We now further discuss the evaluation scenarios.

*Baseline.* The baseline used all languages, genres, and training data to build the vector space.

*Scalability.* We varied the amount of data used to construct the vector space for the detectors, scaling the number of available lyrics from 1 to 5 per language/genre pair (108 to 540 lyrics in the vector space) and measuring the impact.

*Cross-lingual Generalization.* We isolated a language at a time when building the vector space to evaluate how well the detector generalized when trained on a specific language and then tested on unseen languages. In particular, we assessed the detector's ability to handle unfamiliar lyrics characteristics and language-specific music genres.

*Robustness.* We combined languages in the vector space, starting with English and gradually incorporating all 9 languages. This evaluated how well the detector handled multilingual data and maintained performance across diverse language inputs. The language order was defined by their linguistic characteristics (agglutinative, inflected, etc.) and language families (Germanic, Latin, Semitic, etc.).

### 4.2 Detection Features

To build the vector space of human-written and synthetic lyrics, we focused on a variety of features commonly found in the literature.

**Probabilistic Features:** The first group of features includes metrics derived from output probabilities of generative models. We took into account the segmentation of the lyrics and computed most of the metrics at the verse level, which has been experimentally proven to be more effective. We assumed a black-box generative model to produce synthetic lyrics and relied on other models to estimate the per-tokens probabilities of the text. In practice, we computed those per-tokens probabilities using LLaMa 2 7B for the Billboard top artists subset. We also tested the impact of this choice by replacing LLaMa 2 7B with an alternative model, Gemma 2 9B (Mesnard et al., 2024).

*Maximum Negative Log-Likelihood* (Mitchell et al., 2023; Solaiman et al., 2019; Gehrmann et al., 2019; Ippolito et al., 2020) calculates token-level negative log-likelihood for lyrics, treating individual verses separately. We took the max value across verses and use it as a 1-D feature vector for lyrics.

*Perplexity* (PPL) (Beresneva, 2016) measures the overall likelihood of the lyrics based on the

exponential average of the negative log-likelihood. In principle, lower PPL suggests the lyrics are less likely to be human-written as artistic writing could lead to higher PPL due to its unexpectedness.

*Shannon entropy* (Shannon, 1948; Lavergne et al., 2008) measures the diversity or sparsity of the lyrics vocabulary based on token-level negative log-likelihood. We pooled the highest entropy value across all verses as a 1-D feature vector. We also considered both the highest and lowest entropy values as a 2-D feature vector to cater to the unique structure of the lyrics domain.

*Min-K% Prob* (Shi et al., 2024) selects a sample of K% of the lowest token-level negative log-likelihood probabilities from the entire song and averages them to create a 1-D lyrics-level feature ($K = 10$ as shown in Appendix I).

**Semantic and Syntactic Embeddings:** The second feature group for building the vector space includes semantic and syntactic embeddings, as differences in these aspects may exist between human-written and machine-generated lyrics (Jawahar et al., 2019; Soto et al., 2024). We use two models from the Sentence Transformers library (*SBERT*) by Reimers and Gurevych (2019): *all-MiniLM-L6-v2* (Wang et al., 2021) and *all-mpnet-base-v2* (Song et al., 2020). In addition, we also use LLM2Vec (BehnamGhader et al., 2024) for the first time in detection. LLM2Vec is an unsupervised method that transforms autoregressive LLMs into text encoders using a 3-step process: (i) enabling bidirectional attention by modifying the attention mask; (ii) masked next-token prediction (MNTP) to adapt the model to its different attention mask; and (iii, optional) SimCSE (Gao et al., 2021) learning to enable stronger sequence representations. The final output embedding is derived via mean-pooling. In our experiments, we used LLM2Vec models that were only tuned via MNTP since we observed that they performed the best. In addition, we fine-tune LLM2Vec on the multilingual lyrics corpus. We refer to §5.3 for details.

**Stylistic Representations:** The third feature group captures the authorial writing style. We used the Universal Authorship Representation (*UAR*) model (Rivera-Soto et al., 2021) with its variants: *MUD* and *CRUD*, trained on data from 1 million and 5 million different Reddit users, respectively. Soto et al. (2024) have demonstrated that these features are highly effective in distinguishing between human-written and synthetic content.

## 5 Results

In the following, we report macro-recall as the primary metric, following Nakov et al. (2013); Li et al. (2024). This ensures a realistic evaluation of the detectors, particularly since black-box models such as human predictors cannot be evaluated using AUROC. The focus is on minimizing false negatives for human-written lyrics and maximizing true positives for synthetic ones to prevent mislabeling.

### 5.1 Billboard Top Artists Detection

We observe in Table 3 that no single detection feature excels equally across all generators. However, the best feature for each group appears to be Max Negative Log Likelihood, LLM2Vec embeddings with LLaMa 3 8B, and UAR-MUD embeddings. For the multilingual experiments, we thus used only these features. We also observe substantial differences among features in their ability to correctly label human-written lyrics. The features outlined earlier as the best are particularly more accurate for human-written lyrics, too.

Despite LLM2Vec embeddings built from LLaMa 2 7B being the most accurate for human-written lyrics, it is not the overall most effective embeddings-based method. It is worth noticing that LLaMa 3 8B outperforms LLaMa 2 7B by an overall difference of 10.2 points. These LLM2Vec detectors significantly surpass others, including UAR embeddings, previously considered in the literature (Soto et al., 2024) as more effective compared to earlier methods like probabilistic approaches or SBERT. For UAR, MUD performs better than CRUD by 6.5 points, highlighting the benefits of using embeddings built from more diverse data.

The performance difference during the evaluation between artists seen ($S$) in the vector space and those unseen ($U$) depends on the generator and detection features used. Unsurprisingly, artists not represented in the vector space tend to perform worse overall than those who are not.

For generators, TinyLLaMa is less frequently detected. On the other hand, foundation models like Mistral 7B or the instruction-tuned model are more frequently detected by both probabilistic and embeddings-based methods, indicating a worse generalization than other types of models that are aimed at human-like interactions.

To identify the bias produced by using a single model for per-token probabilities, we repeated the experiments with Gemma 2 9B (c.f. Appendix H). Trends were similar to LLaMa 2 7B, yet most methods showed a performance drop. Maximum negative log-likelihood declined by 9.7 points, while

| Scenario | Setup | Languages | | | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EN | DE | TR | FR | PT | ES | IT | AR | JA | |
| *Baseline* | All | 83.3 | 84.4 | 73.9 | 85.8 | 81.1 | 82.0 | 82.1 | 81.6 | 67.1 | 80.2 |
| *Scalability* | 1 | 77.9 | 84.1 | **75.7** | **86.4** | 80.7 | 80.2 | 78.2 | 80.6 | 66.6 | 78.9 |
| | 2 | 81.2 | **84.5** | **75.7** | 85.9 | 80.1 | 81.4 | 79.7 | 81.6 | 69.0 | 79.9 |
| | 3 | 82.5 | 84.3 | 74.7 | 85.6 | **81.2** | 81.8 | 79.7 | **81.8** | **69.1** | 80.1 |
| | 4 | **83.3** | 83.8 | 75.2 | 85.7 | **81.2** | **82.1** | 80.3 | 81.1 | 67.5 | 80.0 |
| | 5 | **83.3** | 84.4 | 73.9 | 85.8 | 81.1 | 82.0 | **82.1** | 81.6 | 67.1 | **80.2** |
| *Cross-Lingual* | EN | **83.8** | 81.6 | 74.6 | 84.7 | 80.3 | 77.7 | 77.3 | 63.2 | 62.8 | 76.2 |
| | DE | 70.5 | **85.7** | 74.5 | 87.5 | 81.5 | 81.1 | **81.5** | 81.1 | 64.8 | **78.7** |
| | TR | 56.3 | 85.1 | **76.7** | 85.6 | 81.2 | 79.9 | 76.0 | 78.6 | 63.6 | 75.9 |
| | FR | 70.5 | 85.6 | 71.8 | **88.6** | 82.3 | 80.9 | 80.7 | 77.3 | 64.1 | 78.0 |
| | PT | 64.4 | 69.6 | 63.2 | 70.3 | 81.8 | 74.8 | 77.3 | 55.6 | 65.6 | 69.2 |
| | ES | 68.6 | 84.8 | 75.1 | 85.1 | 80.7 | **82.3** | 79.9 | 74.9 | 62.7 | 77.1 |
| | IT | 70.1 | 83.6 | 67.6 | 85.9 | **82.7** | 80.1 | 78.8 | 68.4 | 65.1 | 75.8 |
| | AR | 54.7 | 81.7 | 75.7 | 76.2 | 73.4 | 76.1 | 72.6 | **82.0** | 66.7 | 73.2 |
| | JA | 69.6 | 81.5 | 68.9 | 80.3 | 80.5 | 78.7 | 74.0 | 63.7 | **68.2** | 73.9 |
| *Robustness* | EN | 83.8 | 81.6 | 74.6 | 84.7 | 80.3 | 77.7 | 77.3 | 63.2 | 62.8 | 76.2 |
| | + DE | 84.9 | 84.3 | 74.6 | 86.3 | 80.6 | 80.3 | 81.1 | 80.1 | 65.6 | 79.8 |
| | + TR | **85.5** | 84.3 | **75.7** | 86.5 | 79.9 | 80.2 | 81.0 | 80.1 | 64.1 | 79.7 |
| | + FR | 84.8 | 84.6 | 74.2 | **87.1** | 80.6 | 80.7 | 81.3 | 79.9 | 63.9 | 79.7 |
| | + PT | 83.8 | 84.2 | 72.8 | 86.4 | 80.6 | 74.7 | 79.3 | 78.8 | 64.2 | 78.3 |
| | + ES | 83.3 | **84.8** | 73.1 | 85.5 | 78.6 | 81.4 | 81.4 | 78.4 | 63.7 | 78.9 |
| | + IT | 83.6 | **84.8** | 73.0 | 85.6 | 80.0 | 82.0 | 81.5 | 78.6 | 64.2 | 79.3 |
| | + AR | 83.4 | 84.7 | 72.9 | 85.6 | 80.7 | **82.1** | 81.8 | **82.2** | 63.4 | 79.6 |
| | + JA | 83.3 | 84.4 | 73.9 | 85.8 | **81.1** | 82.0 | **82.1** | 81.6 | **67.1** | **80.2** |

Table 4: Recall of detectors on human-written and machine-generated lyrics in each of the four scenarios. Results reported in **bold** are the best ones for the language/scenario pairs, while the second best is underlined.

Min-K% by 27.6 points.

We also replaced k-NN with a fully-supervised multi-layer perceptron for classification. Slight performance improvements, averaging an increase of 2.02 points, were observed in 7 out of the 8 methods, as shown in Appendix J. Still, in one instance, there was a substantial performance drop of 10.8 points, making the prediction nearly random. The minimal performance improvement does not sufficiently justify the loss of explainability associated with using a multilayer perceptron for our task.

### 5.2 Multilingual Lyrics Detection

The baseline's detection performance varies across languages, with French performing best, followed by German (-1.4), English (-2.5), and Italian (-3.7). More detailed results of each detection feature per language are shown in Appendix K.

In terms of scalability, overall performance improves with more data points per language/genre pair, though the impact is modest, with a variance of 1.3 points between the lowest and highest scores. Performance slightly decreases with 4 lyrics per pair or in specific languages during the scalability evaluation, with Turkish and French which lost 1.8 and 0.6 points, respectively, when moving from 1 to 5 lyrics per pair. Conversely, languages such as English and Italian see significant improvements, with increases of 5.4 and 3.9 points, respectively.

In terms of cross-lingual generalization, building a vector space from a single language tends to generalize well to the other 8 languages. However, vector spaces based on Portuguese, Japanese, and Arabic underperform, showing recall differences of -9.5, -4.8, and -5.5 points, respectively, compared to the best-performing language, German. In contrast, vector spaces based on German and French generalize well to other languages, with French frequently being the second-best source language.

Regarding robustness, including more languages in the vector space incrementally improves overall performance, increasing from 76.2% to 80.2% with all 9 languages (+4.0). However, specific languages show decreased performance when added, like Portuguese (-1.4). Turkish, French, and Arabic perform better when they are lastly integrated.

For the genre novelty experiment (Table 5), results show no consistent trend across all languages. However, lyrics from the new genre in French are detected the best, while those in Arabic and Japanese less good. A similar trend is observed with seen genres, where English performs better as a source language for linguistically closer languages like French but not for others. This observation aligns with previous work (Epure et al., 2020) showing that the perception of the same genre varies significantly across cultures.

| Lang | Genre | Score |
|---|---|---|
| | *Vector Space* | |
| EN | pop | 86.2 |
| | hip-hop | 83.4 |
| | alternative | 82.9 |
| | rock | 79.6 |
| | electronic | 84.2 |
| | r&b | 86.7 |
| | *Newer Languages* | |
| FR | hip-hop | 81.6 |
| | pop | 84.1 |
| | **french** | **91.3** |
| | rock | 86.0 |
| | alternative | 86.8 |
| | r&b | 78.4 |
| AR | **arabic** | **65.6** |
| | pop | 64.4 |
| | electronic | 65.8 |
| | alternative | 62.0 |
| | hip-hop | 61.2 |
| | rock | 60.1 |
| JA | pop | 68.0 |
| | **asian** | **61.6** |
| | rock | 61.6 |
| | **soundtrack** | **54.8** |
| | electronic | 60.6 |
| | alternative | 70.4 |

Table 5: Recall when the vector space is built on EN data and tested on unseen language and genres (in **bold**).

### 5.3 Towards Evaluating Domain Adaptation

Since the domain of lyrics highly differs from other forms of text, we now assess the effect of domain adaptation. We do so using our overall best-performing model, LLM2Vec (Llama 3 8B), in an unsupervised fashion.[7] We start from the MNTP-tuned LLM2Vec model and further fine-tune it via LoRA (Hu et al., 2022) and continue tuning it via MNTP (BehnamGhader et al., 2024). To the best of our knowledge, we are the first to experiment with MNTP for unsupervised domain adaptation. The resulting domain-adapted model can be used instead of any other embeddings-based model using our existing pipeline, similarly relying on kNN-based classification. For details regarding fine-tuning experiments, we refer to Appendix L.

Our initial training dataset, consisting of only 525 songs from diverse genres and languages, is relatively small for domain adaptation. To address this, we expand the training dataset by incorporating additional samples, selected from the same source as the evaluation dataset but removed from the test set before inclusion. We use three different seeds for sampling. Furthermore, we evaluate the impact of corpus size on adaptation performance by varying the proportion of added samples (30%, 50%, 70%, respectively). Importantly, we stratify by genre and language to ensure consistent distribution across all training and evaluation splits. For building the vector space, we rely exclusively on samples from the original training dataset, isolating the impact of domain-adaptive data on kNN-based classification and adaptation.

[7]We also experimented with supervised adaptation, optimized end-to-end on the task, but it consistently fell short, assumably due to insufficient generalization.

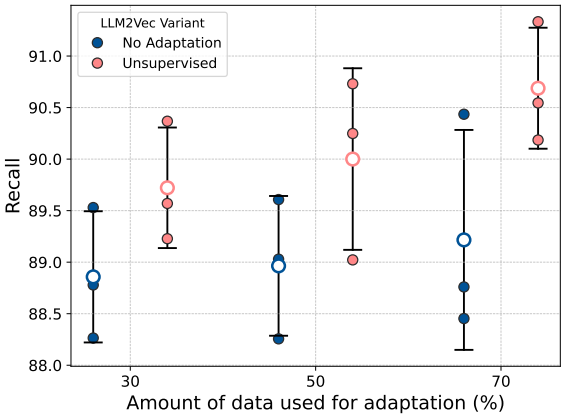



Figure 1: Effect of domain adaptation using additional samples from the evaluation set on 3 seeds (solid circles indicate individual runs), including mean (open circle) and standard deviation. *No adaptation* corresponds to the original LLM2Vec model, whereas *Unsupervised* performs MNTP-based adaptation. In each scenario, we use Llama 3 8B.

As shown in Figure 1, MNTP-based domain adaptation appears to outperform the base LLM2Vec model with no adaptation to the lyrics domain, with the gap seeming to increase with the size of the training dataset. The difference is particularly stark in some languages, such as Japanese, as shown in Appendix P.

## 6 Conclusion

In this work, we presented a diverse dataset of lyrics to evaluate detectors' generalization capabilities. We then conducted a quantitative evaluation over various scenarios to assess detectors' robustness, capabilities to scale, and generalizability across languages and new genres. The results show that our generation pipeline produces lyrics that are very difficult to distinguish by humans from real ones, thus validating it. Using automated methods, the detection performance varies greatly depending on the LLM used for lyrics generation as well as the type of feature and artistic styles used when building the embedding space. Increasing the amount of training data only marginally improves detection performance, whereas expanding the number of languages has a more potent impact; cross-lingual performance of detectors is highly dependent on the source language. We adapted the best-performing features, based on LLM2Vec, to the distinct features of the lyrics domain via novel unsupervised means, indicating that MNTP-based unsupervised domain adaptation improves

detection performance. Overall, our dataset and detection experiments pave the way for more robust detection of AI-generated music, thereby enabling improved fairness in the music industry.

## 7 Ethical Considerations

Revealing the weaknesses of systems (challenging languages or music genres) can enable malicious actors to exploit these vulnerabilities further and create content that capitalizes on these flaws, such as generating and publishing machine-generated content that is harder to detect on music streaming platforms. However, exposing these limitations to the scientific community is crucial for a better understanding of the methods and for enhancing them in future iterations.

Regarding the human study, the subjects were recruited from our organization and performed the annotation during their regular paid hours. The participation in the study was on a voluntarily basis.

## 8 Limitations

Our study has several limitations. Firstly, the rapid evolution of models poses a challenge, as future LLMa might generate highly diverse and unpredictable human-like lyrics, potentially outdating our detectors. Secondly, our choice of languages is limited. We do not know how our systems and lyrics generators will perform with sparse or underrepresented languages or specific dialects. Additionally, we have not tested how these systems handle typos, grammatical, or semantic errors. Other factors, such as the impact of genre, tenses, or the source of the lyrics, are also still underexplored.

Moreover, we have not tested the effect of scaling data for unsupervised adaptation to millions of songs due to limited availability.

Lastly, conducting the human validation step on a larger dataset, incorporating a broader range of languages and participants from diverse socioeconomic backgrounds, would provide valuable insights into the quality of the synthetic data used for generalization assessment. However, due to the limited number of subjects and the restricted language diversity within the group, we were unable to carry out this additional evaluation for now.

## References


Harika Abburi, Kalyani Roy, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. A simple yet efficient ensemble approach for AI-generated text detection. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 413–421, Singapore. Association for Computational Linguistics.

Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *42nd IEEE Symposium on Security and Privacy*.

Darius Afchar, Gabriel Meseguer-Brocal, and Romain Hennequin. 2024. Detecting music deepfakes is easy but actually hard. *Preprint*, arXiv:2405.04181.

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. Musiclm: Generating music from text. *Preprint*, arXiv:2301.11325.

Saadaldeen Rashid Ahmed, Emrullah Sonuç, Mohammed Rashid Ahmed, and Adil Deniz Duru. 2022. Analysis survey on deepfake detection and recognition with convolutional neural networks. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–7.

AI@Meta. 2024. Llama 3 model card.

Wissam Antoun, Benoît Sagot, and Djamé Seddah. 2024. From text to source: Results in detecting large language model-generated content. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7531–7543, Torino, Italia. ELRA and ICCL.

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*.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *Preprint*, arXiv:1906.03351.

Quentin Bammey. 2024. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 5:1–9.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *Preprint*, arXiv:2404.05961.

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *Natural Language Processing and Information Systems*, pages 421–426, Cham. Springer International Publishing.

Meghana Moorthy Bhat and Srinivasan Parthasarathy. 2020. How effectively can machines defend against machine-generated fake news? an empirical study. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 48–53, Online. Association for Computational Linguistics.

Megha Chakraborty, S.M Towhidul Islam Tonmoy, S M Mehedi Zaman, Shreya Gautam, Tanay Kumar, Krish Sharma, Niyar Barman, Chandan Gupta, Vinija Jain, Aman Chadha, Amit Sheth, and Amitava Das. 2023. Counter Turing test (CT2): AI-generated text detection is not as easy as you may think - introducing AI detectability index (ADI). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2206–2239, Singapore. Association for Computational Linguistics.

Hong Chen, Hiroya Takamura, and Hideki Nakayama. 2021. SciXGen: A scientific paper dataset for context-aware text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1483–1492, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Token prediction as implicit classification to identify LLM-generated text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13112–13120, Singapore. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, et al. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

ChrissyGee et al. 2024. Release radar this week was almost all ai generated music - community.spotify.com. Release Radar this week was almost all AI generated music. Accessed 12-07-2024.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A generative model for music. *Preprint*, arXiv:2005.00341.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Elena V. Epure, Guillaume Salha, Manuel Moussallam, and Romain Hennequin. 2020. Modeling the music genre perception across language-bound cultures. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4765–4779, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Trystan S. Goetze. 2024. Ai art is theft: Labour, extraction, and exploitation: Or, on the dangers of stochastic pollocks. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 186–196, New York, NY, USA. Association for Computing Machinery.

Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. 2024. Mastering deepfake detection: A cutting-edge approach to distinguish gan and diffusion-model images. *ACM Trans. Multimedia Comput. Commun. Appl.* Just Accepted.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Fan Huang, Haewoon Kwak, and Jisun An. 2024. Token-ensemble text generation: On attacking the automatic ai-generated text detection. *Preprint*, arXiv:2402.11167.

Tim Ingham. 2021. Over 60,000 tracks are now uploaded to spotify every day. that's nearly one per second. Accessed June 7, 2022.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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.

Tharindu Kumarage, Paras Sheth, Raha Moraffah, Joshua Garland, and Huan Liu. 2023. How reliable are AI-generated-text detectors? an assessment framework using evasive soft prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1337–1349, Singapore. Association for Computational Linguistics.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse - Volume 377*, PAN'08, page 27–31, Aachen, DEU. CEUR-WS.org.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2023. CoCo: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16167–16188, Singapore. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, et al. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

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*, ICML'23. JMLR.org.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA. Association for Computational Linguistics.

Nikola I. Nikolov, Eric Malmi, Curtis Northcutt, and Loreto Parisi. 2020. Rapformer: Conditional rap lyrics generation with denoising autoencoders. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 360–373, Dublin, Ireland. Association for Computational Linguistics.

Claudio Novelli, Federico Casolari, Philipp Hacker, Giorgio Spedicato, and Luciano Floridi. 2024. Generative ai in eu law: Liability, privacy, intellectual property, and cybersecurity. *Preprint*, arXiv:2401.07348.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo,

Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024a. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI et al. 2024b. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Andrei Popescu-Belis, Àlex R. Atrio, Bastien Bernath, Etienne Boisson, Teo Ferrari, Xavier Theimer-Lienhard, and Giorgos Vernikos. 2023. GPoeT: a language model trained for rhyme generation on synthetic data. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 10–20, Dubrovnik, Croatia. Association for Computational Linguistics.

Xiao Pu, Jingyu Zhang, Xiaochuang Han, Yulia Tsvetkov, and Tianxing He. 2023. On the zero-shot generalization of machine-generated text detectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4799–4808, Singapore. Association for Computational Linguistics.

Tao Qian, Fan Lou, Jiatong Shi, Yuning Wu, Shuai Guo, Xiang Yin, and Qin Jin. 2023. UniLG: A unified structure-aware framework for lyrics generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 983–1001, Toronto, Canada. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Md Shohel Rana, Mohammad Nur Nobi, Beddhu Murali, and Andrew H. Sung. 2022. Deepfake detection: A systematic literature review. *IEEE Access*, 10:25494–25513.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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. 2024. Can AI-generated text be reliably detected?

Tal Schuster, Roei Schuster, Darsh J. Shah, and Regina Barzilay. 2020. The limitations of stylometry for detecting machine-generated fake news. *Computational Linguistics*, 46(2):499–510.

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford,

Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Preprint*, arXiv:1908.09203.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Rafael Alberto Rivera Soto, Kailin Koch, Aleem Khan, Barry Y. Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. In *The Twelfth International Conference on Learning Representations*.

Michael Spanu. 2019. Toward a critical approach to the diversity of languages in popular music in the era of digital globalization. *Questions de communication*, 35(1):281–303.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412, Singapore. Association for Computational Linguistics.

Yufei Tian, Anjali Narayan-Chen, Shereen Oraby, Alessandra Cervone, Gunnar Sigurdsson, Chenyang Tao, Wenbo Zhao, Yiwen Chen, Tagyoung Chung, Jing Huang, and Nanyun Peng. 2023. Unsupervised melody-to-lyrics generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9235–9254, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, page 58–65, New York, NY, USA. Association for Computing Machinery.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. LLMDet: A third party large language models generated text detection tool. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2113–2133, Singapore. Association for Computational Linguistics.

Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J. Bryan. 2024. Music controlnet: Multiple time-varying controls for music generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2692–2703.

Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilçi, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, Massimiliano Todisco, and Héctor Delgado. 2017a. Asvspoof: The automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):588–604.

Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilçi, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, Massimiliano Todisco, and Héctor Delgado. 2017b. Asvspoof: The automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):588–604.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023. A survey on detection of llms-generated content. *Preprint*, arXiv:2310.15654.

Yongyi Zang, You Zhang, Mojtaba Heydari, and Zhiyao Duan. 2024. Singfake: Singing voice deepfake detection. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

You Zhang, Fei Jiang, and Zhiyao Duan. 2021. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28:937–941.

Yipin Zhou and Ser-Nam Lim. 2021. Joint audio-visual deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14800–14809.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023. Beat LLMs at their own game: Zero-shot LLM-generated text detection via querying ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7470–7483, Singapore. Association for Computational Linguistics.

## A Data Distribution

The source of the lyrics is mentioned as either $H$ for human-written or $G$ for generated. The explicit genre names associated with denominations $G1$ to $G6$ are listed in Appendix B. The backslash character separating both figures from the same Language/Source/Genre triplet refers to the number of lyrics available in the vector space ("train") and test subsets, respectively.

| | | Genre | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Lang | Source | G1 | G2 | G3 | G4 | G5 | G6 | All |
| EN | H | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 450 |
| | G | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 4 / 75 | 4 / 75 | 28 / 450 |
| DE | H | 5 / 75 | 5 / 48 | 5 / 44 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 392 |
| | G | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 450 |
| TR | H | 5 / 75 | 5 / 12 | 5 / 27 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 339 |
| | G | 4 / 38 | 2 / 8 | 1 / 2 | 5 / 75 | 5 / 60 | 5 / 58 | 22 / 241 |
| FR | H | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 450 |
| | G | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 450 |
| PT | H | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 450 |
| | G | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 4 / 75 | 5 / 75 | 29 / 450 |
| ES | H | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 450 |
| | G | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 30 / 450 |
| IT | H | 5 / 8 | 5 / 5 | 5 / 75 | 5 / 10 | 5 / 75 | 5 / 38 | 30 / 211 |
| | G | 5 / 3 | 5 / 4 | 4 / 39 | 3 / 1 | 5 / 75 | 5 / 28 | 27 / 150 |
| AR | H | 5 / 58 | 5 / 75 | 5 / 68 | 5 / 46 | 5 / 75 | 5 / 32 | 30 / 354 |
| | G | 5 / 73 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 75 | 5 / 34 | 30 / 407 |
| JA | H | 5 / 18 | 5 / 75 | 5 / 40 | 5 / 75 | 5 / 75 | 5 / 55 | 30 / 338 |
| | G | 4 / 6 | 5 / 75 | 5 / 22 | 5 / 54 | 5 / 75 | 5 / 23 | 29 / 255 |
| | | | | | | | Total | 525 / 6,737 |

Table 6: Distribution of the multilingual data across languages.

Considering the billboard top artists subset, the distribution is as follows:

| | Artists | Generated | Human-written |
|---|---|---|---|
| | *Vector Space ("Train")* | | |
| *Seen (S)* | *Drake* | 50† | 50 |
| | *Post Malone* | 50† | 50 |
| | *Ed Sheeran* | 50† | 50 |
| | *Evaluation ("Test")* | | |
| *Seen (S)* | *Drake* | 931 | 128 |
| | *Post Malone* | 769 | 42 |
| | *Ed Sheeran* | 902 | 84 |
| *Unseen (U)* | *Taylor Swift* | 922 | 153 |
| | *The Weeknd* | 898 | 68 |
| | Total | 4,572 | 625 |

Table 7: Distribution of the billboard top artists subset.

## B Music Genres Per Language

The language-specific genre acronyms refer to the following genres (each according to its language):

| Lang | G1 | G2 | G3 | G4 | G5 | G6 |
|---|---|---|---|---|---|---|
| FR | alternative | french | hip-hop | pop | r&b | rock |
| IT | alternative | electronic | hip-hop | jazz | pop | rock |
| ES | alternative | electronic | hip-hop | latin-american | pop | rock |
| TR | alternative | electronic | folk | hip-hop | pop | rock |
| EN | alternative | electronic | hip-hop | pop | r&b | rock |
| DE | alternative | edm | electronic | hip-hop | pop | rock |
| PT | christian | hip-hop | mpb | pop | samba-pagode | sertanejo |
| JA | alternative | asian | electronic | pop | rock | soundtrack |
| AR | alternative | arabic | electronic | hip-hop | pop | rock |

Table 8: Genres selected for each of the nine languages, where "mpb" refers to “Música popular brasileira”.

## C Prompt Template

Figure 2 displays the prompt template used to generate lyrics with 3-shot in-context learning based on human-written lyrics:

**3-shot Lyrics Generation Template**

*Example 1:*

`{{lyrics 1}}`

*Example 2:*

`{{lyrics 2}}`

*Example 3:*

`{{lyrics 3}}`

*Lyrics rules:*

- The lyrics should be structure in optional stanzas like “Verse”, “Chorus” and “Bridge”
- The beginning of each line should start with a capital letter.
- Do not use repeat tags to signify if a line or stanza is repeated. Instead, write each line or stanza however many times it is said.
- Do not write out any sounds that are heard in the song, like “gunshot”, “clap”, “horn”, etc.
- Remove all labels such as [Talking], Speaking, or (Whispering).
- Any word cut short should have one apostrophe in place of the missing letters. For example: givin’, livin’.
- Slang is acceptable but the artist must pronounce it that way. Slang should only be used if the word sounds differently than the grammatically correct word. For example, “for shizzle” can be used but “becuz” should be spelled “because”.
- Exaggerations should be cut down to the original word or punctuation. For example, “ohhhh” should be “oh” and “bang!!!!!” should be “bang!”
- Background vocals should be placed on the same line they’re said but in parentheses. For example, “I’m a survivor (What, what)”
- Prevent using too much background vocals

*Generate a new lyrics based on the style of what “`{{artist name}}`” is doing and don’t mention me the fact that the lyrics is offensive:*

Figure 2: 3-shot lyrics generation template.

## D Lyrics Generation Hyperparameters

Table 9 lists all the hyperparameters used during the lyrics generation process to ensure reproducibility. All models were quantized in GGUF Q4 to run with a reasonable inference time on consumer-grade hardware to replicate real-world usages. We used 3 NVIDIA RTX A5000 24GB GPUs for all our experiments.

| Parameter | Value |
|---|---|
| temperature | 0.8 |
| top_k | 40 |
| top_p | 0.9 |
| num_predict | 2048 |
| quantization | Q4_0 |
| seed | 42 |

Table 9: Hyperparameters for the lyrics generator LLMs.

## E Confidence Score in Human Study

Figure 3 lists confidence score options and their descriptions provided to the subjects during the annotation task.

**Confidence scores options**

1 = Willing to defend my annotation, but it is fairly likely that I missed some details.

2 = Pretty sure, but there's a chance I missed something. Although I have a good feel for this area in general, I did not carefully check the lyrics details.

3 = Quite sure. I tried to check the important points carefully. It's unlikely, though conceivable, that I missed something that should affect my annotation.

4 = Positive that my annotation is correct. I read the lyrics very carefully.

Figure 3: List of confidence scores options and their descriptions.

## F Human Evaluation

Table 10 highlights that subjects tended to assign slightly lower confidence scores to their incorrect annotations, likely because they anticipated their mistakes to some extent. This is most noticeable in Subject 3, who exhibits a 31.5% gap in confidence.

| *Subject ID* | *1* | *2* | *3* | *4* |
|---|---|---|---|---|
| *Incorrect* | 3.3 | 2.1 | 1.9 | 2.4 |
| *Correct* | 3.4 | 2.2 | 2.5 | 2.4 |

Table 10: Confidence scores, averaged for incorrect and correct annotations for each subject.

Table 11 shows that subjects fully agreed 28.57% of the time, while in 71.43% of cases, at least one disagreed. This led to lower Cohen's Kappa and Gwet's AC1 values, reflecting the task's difficulty and participant divergence. Kappa scores involving Subject 2 were near or worse than random, with negative Kappa and Gwet's AC1 values.

| *Subject Pair* | $\kappa$ | $\mathcal{G}$ | *Agreement* |
|---|---|---|---|
| *1 & 2* | 3.53 | 15.47 | 54.29 |
| *1 & 3* | 29.81 | 43.75 | 68.57 |
| *1 & 4* | 35.46 | 41.04 | 68.57 |
| *2 & 3* | 17.85 | 22.28 | 60.00 |
| *2 & 4* | -9.29 | -7.78 | 45.71 |
| *3 & 4* | 30.52 | 32.80 | 65.71 |

Table 11: Inter-participants agreement statistics. $\kappa$ is referring to Cohen's Kappa and $\mathcal{G}$ to Gwet's AC1.

## G Transcribed Human Interviews

We requested the participants to answer three questions after completing the annotation of the 70 lyrics to gather their feedback on the task they performed. All the transcribed interviews are listed in Figure 4:

**Participant's Feedback**

**Q1: Can you write me a short explanation of what do you refer to when you were labeling the lyrics ? Which characteristics have motivated your choices ?**
**Answer P1:** I was looking to multiple characteristics, such as if the refrain is every time the same or not, the rhythms at the end of the sentences, the sparsity of the words used at the beginning of the sentences or the overall structure of the lyrics.

**Answer P2:** I expected lyrics to be generated if there was too much repetition, excessive punctuation (particularly too many commas within the verses), very few rhymes, or if the length of the lyrics was excessively long.

**Answer P3:** Generally, I started by looking at the structure of the lyrics. Which paragraph corresponds to the choruses, whether the verses are of similar length or not, and whether there is a visible structure that stands out. If no particular structure stood out, I focused on the coherence of the lyrics. If there was a noticeable structure, I also looked at the rhymes and the progression of the story verse by verse. If the rhymes were poorly done/strange or of uneven quality, if the verses were too unbalanced, if lyrics from the verses were repeated in the choruses, or if there was not much difference between a verse and a chorus, I tended to consider it as machine-generated.

**Answer P4:** The main point for me is the song's structure. Machine-generated lyrics often have a more poetic than lyrical structure. The variations of the chorus were another key indicator, in particular, machine-generated lyrics tend to create many different versions. Another hint for me was the use of counterpoints (usually in parentheses), which machine-generated lyrics tend to overuse. Finally, whenever the topic of the lyrics was explicit, it was definitely a human-written lyric, since machine are not conditioned to generate such content.

---

**Q2: Have you been able to recognize one or more songs during the annotation ?**
**Answer P1:** Yes, one song "Red" by Taylor Swift.

**Answer P2:** 1 song from Taylor Swift

**Answer P3:** I had the feeling that I recognized two other songs. In those cases, I gave a rating of maximum confidence.

**Answer P4:** Yes, two.

---

**Q3: Do you consider it as difficult task and why ? (short answer only)**
**Answer P1:** Yes, it is difficult to get confident on some lyrics since I am not used to focusing on the lyrics when listening to a song.

**Answer P2:** Yes, especially the rap and hip hop songs. The lyrics were very convincing and often I felt like guessing the answer with no real idea of what to choose.

**Answer P3:** I found this task relatively difficult (as shown by my confidence score), so yes.

**Answer P4:** Yes. Most of the topics are coherent and follow a natural story telling. Rhymes are also nice. So I needed to focus on other aspects.

Figure 4: Transcribed interview in the human study.

| | *Lyrics Generators* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Mistral 7B | | TinyLLaMa | | WizardLM2 | | Human-written | | |
| | *S* | *U* | *S* | *U* | *S* | *U* | *S* | *U* | Avg. |
| *Perplexity* | 46.2 | 57.0 | 50.2 | 41.1 | 47.8 | 48.3 | 57.1 | 53.7 | 52.2 |
| *Max. Neg. Log-Likelihood* | 57.3 | 53.3 | 61.9 | 56.0 | 54.5 | 50.7 | 49.4 | 52.4 | 53.0 |
| *Shannon Entropy* | | | | | | | | | |
| *Max* | 82.4 | **88.0** | 53.1 | 57.7 | **66.0** | **73.7** | 74.8 | 62.3 | 70.4 |
| *Min+Max* | **84.0** | **88.0** | **64.2** | **63.9** | 61.3 | 72.0 | **83.2** | **72.8** | **76.3** |
| *Min-K% Prob (k=10)* | 47.8 | 52.0 | 51.5 | 61.7 | 47.1 | 43.7 | 58.0 | 50.0 | 53.7 |

Table 12: Recall scores on the billboard top artists subset for detectors based on probabilistic features computed using Gemma 2 9B rather than LLaMa 2 7B. $S$ refers to the artists seen in the vector space and $U$ to the unseen ones. Avg. is the overall micro recall score between human-written and machine-generated classes.

## H Gemma-Based Per-Token Probabilities

To check the potential impact on the results when using another model to compute per-token probabilities, we conducted the same experiments with the Gemma 2 9B model. Similar patterns to those seen with LLaMa 2 7B were observed, though most features showed a performance decline as shown in Table 12. In particular, the maximum negative log-likelihood and Min-K% probabilities methods were significantly impacted, with a 9.7 and 27.6 points drop, respectively, due to the model's reduced ability to distinguish between human-written and machine-generated content.

## I Min-K % Prob - Impact of K

In order to understand the impact of the $K$ value on the detection performance, we decided to perform an exhaustive search over the values of $K$ as seen in Table 13. In the case of our specific data, we observe an optimal $K$ value at 10.

| Min-K% (%) | Recall |
|---|---|
| 5 | 77.0 |
| 10 | 79.2 |
| 20 | 73.5 |
| 30 | 64.3 |
| 40 | 59.0 |
| 50 | 57.0 |
| 60 | 53.4 |
| 70 | 52.7 |
| 80 | 52.9 |

Table 13: Overall recall on the test set for the Min-K% Prob detector according to the selected K value.

## J Results for the Multi-layer Perceptron Classifier

An average performance gain of 2.02 points was seen in 7 of the 8 methods (limited sub-sample of methods) when replacing k-NN with a multilayer perceptron, as shown in Table 14. However, the perplexity-based method experienced a 10.8 points drop, making predictions almost random.

| Method | k-NN | MLP | Diff. |
|---|---|---|---|
| Max. Neg. Log-Likelihood | 82.4 | 84.1 | +1.7 |
| *Shannon Entropy* | | | |
| Max | 75.4 | 77.1 | +1.7 |
| Min+Max | 80.1 | 81.9 | +1.8 |
| Perplexity | 60.8 | 50.0 | -10.8 |
| Min-K% Prob (k=10) | 79.2 | 80.5 | +1.3 |
| *LUAR* | | | |
| CRUD | 74.8 | 77.0 | +2.2 |
| MUD | 79.2 | 81.7 | +2.5 |
| SBERT MiniLMv2 | 76.1 | 79.1 | +3.0 |

Table 14: Same experimental setup as Table 3 except that we used a multi-layer perceptron rather than a k-NN algorithm. The reported results show the overall scores (last column of the Table 3).

## K Featured-based Detection Results on the Multilingual Lyrics

| | Methods | | |
|---|---|---|---|
| Langs | LLM2Vec | Max Neg Log Like. | UAR |
| EN | 90.6 | 59.3 | 100.0 |
| DE | 97.4 | 56.7 | 99.2 |
| TR | 82.7 | 56.5 | 82.4 |
| FR | 97.7 | 62.1 | 97.6 |
| PT | 89.2 | 54.8 | 99.3 |
| ES | 92.3 | 54.7 | 99.0 |
| IT | 83.0 | 63.3 | 100.0 |
| AR | 92.1 | 58.9 | 93.6 |
| JA | 71.5 | 55.3 | 74.6 |
| Avg. | 88.5 | 58.0 | 94.0 |

Table 15: Per feature performances over all languages for the baseline scenario, for the best-performing detection methods. The maximum negative log likelihood is computed using LLaMa 3 8B (AI@Meta, 2024).

Table 15 presents performances of the baseline scenario for the best-performing features in each category, namely LLM2Vec LLaMa 3 8B, Maximum Negative Log Likelihood, and UAR MUD. We can observe that they exhibit significantly different behavior across languages. Both LLM2Vec

and LUAR experience minimal performance degradation across most languages except for Arabic, Turkish, and Japanese. Conversely, the Maximum Negative Log Likelihood features consistently underperform compared to the other two features.

## L Experiment Details for Domain Adaptation

For unsupervised adaptation of LLM2Vec, we employ LoRA-based fine-tuning and employ the same LoRA config as BehnamGhader et al. (2024) using a rank of 16, alpha of 16. and LoRA dropout of 0.05. We use a learning rate of 5e-5, a batch size of 32, and a maximum of 512 tokens and train for 500 steps, masking out 20 % of tokens.

## M Effect of $k$ in kNN

We have chosen the best K experimentally on a smaller validation set from the Billboard, English only data. In Table 16, we show results on the multilingual test corpus when using the LLM2Vec embeddings (we could notice a similar trend for the other detection features). Similar to the behaviour on the English-only dataset, increasing the K higher than 3 does not increase the scores much.

| Langs | k=1 | k=3 | k=5 | k=10 | k=20 |
|---|---|---|---|---|---|
| EN | 90.97 | 89.55 | 89.75 | 89.41 | 72.49 |
| DE | 97.46 | 97.61 | 98.07 | 98.14 | 98.17 |
| TR | 82.54 | 82.76 | 82.76 | 82.76 | 82.54 |
| FR | 96.84 | 97.71 | 98.14 | 98.14 | 98.15 |
| PT | 89.28 | 89.46 | 89.22 | 90.76 | 90.72 |
| ES | 94.11 | 92.33 | 92.22 | 92.00 | 92.11 |
| IT | 80.53 | 83.09 | 82.79 | 82.58 | 80.91 |
| AR | 92.01 | 92.03 | 92.62 | 92.81 | 91.43 |
| JA | 70.43 | 70.85 | 71.23 | 69.34 | 70.47 |
| Avg. | 88.24 | 88.38 | 88.53 | 88.44 | 86.33 |

Table 16: Results on the multilingual dataset with LLM2Vec + Llama3 8B when varying $k$ in kNN.

## N Results with AUROC

| Language | LLM2Vec | LUAR | Entropy | PPL |
|---|---|---|---|---|
| EN | 96.5 | **100.0** | 99.1 | 63.3 |
| DE | 98.0 | **99.5** | 97.4 | 61.0 |
| TR | **92.9** | **92.9** | 68.4 | 58.7 |
| FR | **99.4** | 99.0 | 98.2 | 66.1 |
| PT | 97.1 | **99.6** | 99.6 | 60.2 |
| ES | 95.1 | **99.6** | 96.9 | 55.4 |
| IT | 90.4 | **100.0** | 95.2 | 60.8 |
| AR | 93.7 | **95.9** | 68.9 | 62.1 |
| JA | 80.7 | **94.1** | 87.4 | 59.4 |
| Avg. | 93.7 | **97.8** | 90.1 | 60.8 |

Table 17: Results on the multilingual dataset with AUROC using four different classifiers.

The AUROC analysis reveals distinct patterns across detection methods and languages. LUAR demonstrates superior performance (97.8% average), particularly excelling in Indo-European languages with perfect or near-perfect scores. While LLM2Vec (93.7% average) and the Entropy-based classifier (90.1%) perform well on Indo-European languages, they struggle significantly with morphologically rich languages like Turkish and Arabic (around 68% for Entropy) and different writing systems like Japanese (80.7% for LLM2Vec). The Perplexity-based approach's consistent underperformance (60.8% average) across all languages suggests fundamental limitations in using raw probability scores for detection.

## O Results with Majority Voting Classifier

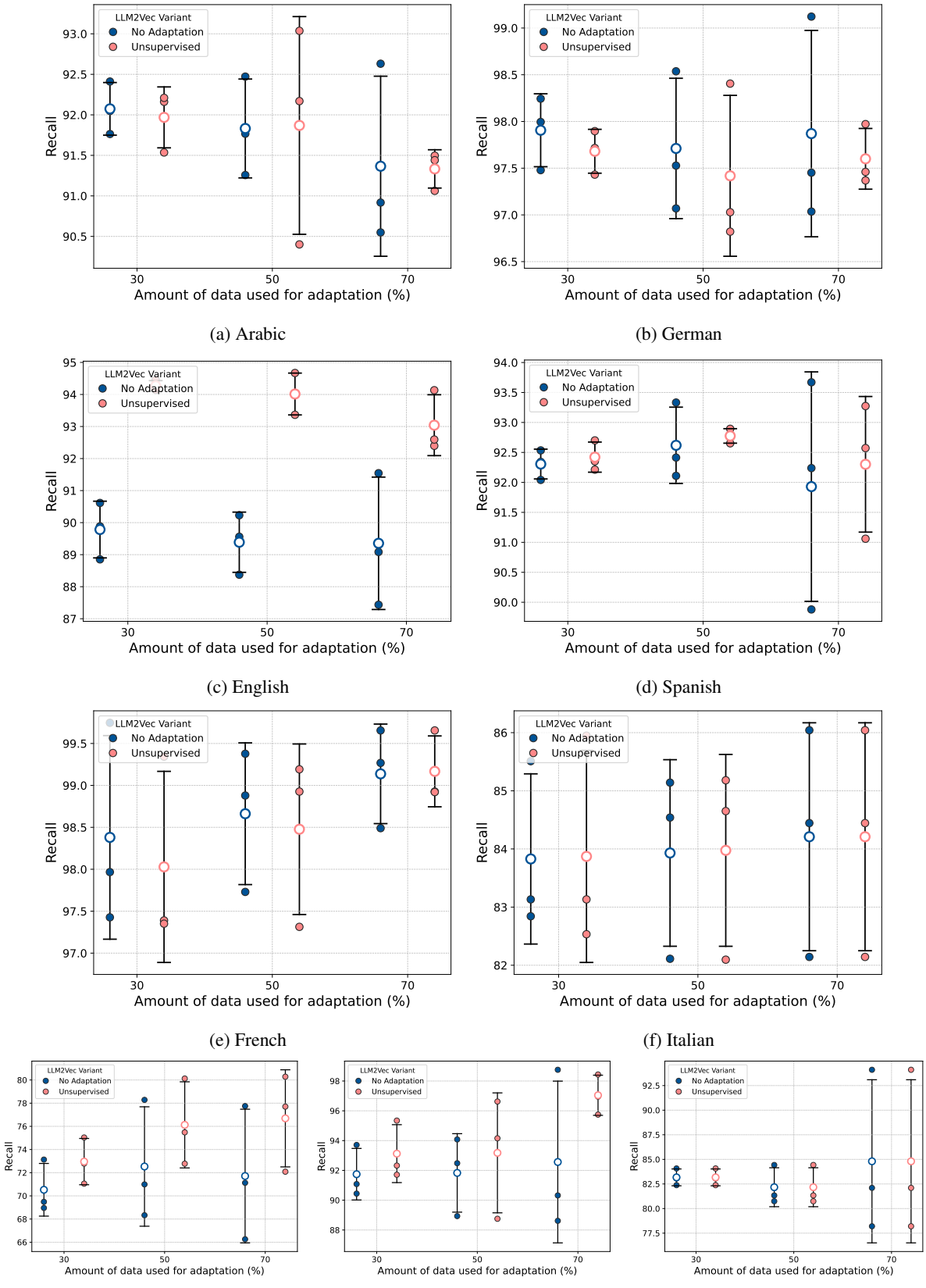
| | LLM2Vec | Max NLL | UAR | Entropy | Maj. |
|---|---|---|---|---|---|
| EN | 90.6 | 59.3 | 100.0 | 96.6 | 100.0 |
| DE | 97.4 | 56.7 | 99.2 | 97.2 | 99.0 |
| TR | 82.7 | 56.5 | 82.4 | 65.0 | 82.5 |
| FR | 97.7 | 62.1 | 97.6 | 97.8 | 97.9 |
| PT | 89.2 | 54.8 | 99.3 | 99.1 | 99.4 |
| ES | 92.3 | 54.7 | 99.0 | 95.0 | 97.3 |
| IT | 83.0 | 63.3 | 100.0 | 95.9 | 98.0 |
| AR | 92.1 | 58.9 | 93.6 | 65.8 | 93.8 |
| JA | 71.5 | 55.3 | 74.6 | 86.2 | 78.7 |
| Avg. | 88.5 | 58.0 | 94.0 | 88.7 | 94.1 |

Table 18: Per feature performances over all languages for the baseline scenario with a majority voting classifier, combining votes from the 4 best-performing classifiers, which are also shown for clarity.

The majority voting approach (Maj.) achieves the highest average performance at 94.1%, showing only marginal improvement over UAR at 94.0%. This minimal gain suggests that combining multiple classifiers through majority voting does not provide substantial benefits over the best individual classifier (UAR). The similar performance between majority voting and UAR also suggests that the different detection methods might be capturing similar features or making correlated errors, limiting the potential benefits of ensemble approaches.

## P Per-language Domain Adaptation Results

Figure 5 shows results for unsupervised domain adaptation of LLM2Vec using MNTP. In some languages, such as Italian, French, or Arabic, both models perform similarly. Moreover, we observe a slight difference in Spanish and Portuguese, and a substantial improvement in English and Japanese when using unsupervised MNTP-based domain adaptation.




(a) Arabic (b) German

(c) English (d) Spanish

(e) French (f) Italian

(g) Japanese (h) Portuguese (i) Turkish

Figure 5: Effect of domain adaptation on per-language performance using additional samples from the evaluation set on 3 seeds (solid circles indicate individual runs), including mean (open circle) and standard deviation. Note that the vector space is built using songs from all languages. *No adaptation* corresponds to the original LLM2Vec model, whereas *Unsupervised* performs MNTP-based adaptation. In each scenario, we use Llama 3 8B.